2003-09

Interoperability, data control and battlespace visualization using XML, XSLT and X3D

Neushul, James D.

Monterey, California. Naval Postgraduate School

NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

INTEROPERABILITY, DATA CONTROL AND BATTLESPACE VISUALIZATION USING XML, XSLT AND X3D

by

James D. Neushul

September 2003

Thesis Advisor: Don Brutzman
Second Reader: Curtis L. Blais

THIS PAGE INTENTIONALLY LEFT BLANK

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	3. REPORT TYPE AND DATES COVERED
	September 2003	Master's Thesis

4. TITLE AND SUBTITLE
Interoperability, Data Control and Battlespace Visualization using XML, XSLT and X3D

5. FUNDING NUMBERS

6. AUTHOR
James D. Neushul

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)
Space and Naval Warfare Systems Center, San Diego California (SSC San Diego)

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
A

13. ABSTRACT

This work represents the realization of Network-Centric goals of interoperability, information management, systems integration and cohesive battlespace visualization using networked computer technology. The application of structured data methodologies using the Extensible Markup Language (XML) allows organizations and systems to exchange and process battlespace information cooperatively. The practical application of this technology is demonstrated.

Governance of information systems using structured data and the rejection of proprietary, application specific solutions is a leadership responsibility that is defined as Data Control. XML is presented as a leadership control measure that can be used to achieve Network-Centricity on the battlefield.

The fundamental principles of XML application development are presented in the context of warfighting. Exemplars address a cross-section of battlespace applications. The visualization of the physical battlefield is demonstrated with network delivered 3D terrain views. Geodesy and position reporting is addressed using an XML defined data structure to enforce interoperability. An XML expression of the Battlespace Generic Hub is applied to joint and multilateral interoperability and information exchange. An approach to the effective employment of multiple different, but cooperative, autonomous systems in the battlespace uses XML to define parameters that determine artificial intelligence multi-agent behavior and environmental factors.

This thesis combines a critical analysis of the priorities of Network-Centricity and interoperability with practical and functional exemplars that demonstrate the efficacy of extensible architectures. The pragmatic approach is directed at the warfighter, and leadership challenges are identified.

14. SUBJECT TERMS
Extensible Markup Language, XML, Extensible 3D Graphics, X3D, Extensible Stylesheet Language for Transformation, XSLT, XML Schema, Interoperability, Battlespace Visualization, Command and Control, Data Control, Digital Terrain Elevation Data(DTED), Geodesy, Position Reporting, Unmanned Aerial Vehicle (UAV), Multi-Agent Systems, Battlespace Generic Hub, Land Command and Control Information Exchange Data Model, LC2IEDM, ATCCIS, Network-Centric Warfare, Java, Extensible Modeling and Simulation Framework, XMSF, MOVES, SAVAGE

15. NUMBER OF PAGES
197

16. PRICE CODE

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

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

INTEROPERABILITY, DATA CONTROL AND BATTLESPACE VISUALIZATION USING XML, XSLT AND X3D

James D. Neushul
Captain, United States Marine Corps
B.A., University of California at Santa Barbara, 1996

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author: James D. Neushul

Approved by: Don Brutzman
Thesis Advisor

Curtis L. Blais
Second Reader/Co-Advisor

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This work represents the realization of Network-Centric goals of interoperability, information management, systems integration and cohesive battlespace visualization using networked computer technology. The application of structured data methodologies using the Extensible Markup Language (XML) allows organizations and systems to exchange and process battlespace information cooperatively. The practical application of this technology is demonstrated.

Governance of information systems using structured data and the rejection of proprietary, application specific solutions is a leadership responsibility that is defined as Data Control. XML is presented as a leadership control measure that can be used to achieve Network-Centricity on the battlefield.

The fundamental principles of XML application development are presented in the context of warfighting. Exemplars address a cross-section of battlespace applications. The visualization of the physical battlefield is demonstrated with network delivered 3D terrain views. Geodesy and position reporting is addressed using an XML defined data structure to enforce interoperability. An XML expression of the Battlespace Generic Hub is applied to joint and multilateral interoperability and information exchange. An approach to the effective employment of multiple different, but cooperative, autonomous systems in the battlespace uses XML to define parameters that determine artificial intelligence multi-agent behavior and environmental factors.

This thesis combines a critical analysis of the priorities of Network-Centricity and interoperability with practical and functional exemplars that demonstrate the efficacy of extensible architectures. The pragmatic approach is directed at the warfighter, and leadership challenges are identified.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
ARM	ATTCIS Replication Mechanism
ATCCIS	Army Tactical Command and Control Information System
BAA	Office of Naval Research Broad Area Announcement
BFT	Blue Force Tracker
BGH-ARM	ATTCIS Replication Model
BIEML	Battlespace Information Exchange Markup Language
C/JMTK	Commercial Joint Mapping Toolkit
C2	Command and Control
C2IEDM	Command and Control Information Exchange Data Model
C2IEDM	Command and Control Information Exchange Data Model
C2IS	Command and Control Information Systems
C4I3	Command, Control, Computers, Communication, Information, Intelligence and Interoperability
COBP	NATO Code of Best Practice
COE	Common Operating Environment
COM	Common Object Model
COTS	Commercial-Off-The-Shelf
DCEE	Distributed Continuous Experimentation Exercise
DED	Digital Elevation Data
DEM	Digital Elevation Map

DII-COE	Defense Information Infrastructure Common Operating Environment
DOM	Document Object Model
DTED	Digital Terrain Elevation Data
FRD	Functional Requirements Document
GIG	Global Information Grid
GML	Geography Markup language
GOTS	Government-Off-The-Shelf
GPP	Generic Positioning Protocol
GTOPO	Geographic Topological Data
GUI	Graphical User Interface
HLA	High Level Architecture
HTML	Hypertext Markup Language
IDA	Institute for Defense Analysis
ISO	International Standards Organization
JCATS	Joint Conflict and Tactical Simulation
JMTK	Joint Mapping Toolkit
LIMDIS	Limited Distribution
M&S	Modeling and Simulation
MAS	Multi-Agent Systems
MDACT	Mobile Data Communication Terminal
MGRS	Military Grid Reference System
MIP	Multilateral Interoperability Programme

NCES	Net Centric Enterprise Services
NIMA	National Imagery and Mapping Agency
NMEA-108	National Marine Electronics Association specification
OIF	Operation Iraqi Freedom
PRL	Position Reporting Language
QP	Qualitative Physics
QSIM	Qualitative Simulation
SAVAGE	Scenario Authoring and Visualization for Advanced Graphical Environments
SQL	Structured Query Language
SVG	Scalable Vector Graphics
UAAV	Unmanned Autonomous Aerial Vehicle
UAV	Unmanned Aerial Vehicle
USGS	United States Geological Survey
UTM	Universal Transverse Mercator
VRML	Virtual Reality Modeling Language
W3C	World Wide Web Consortium
WWW	World Wide Web
X3D	Extensible 3D Graphics Language
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XMSF	Extensible Modeling and Simulation Framework
XSLT	Extensible Stylesheet Language for Transformation

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to acknowledge the direct financial support of the Space and Naval Warfare Systems Center, San Diego California (SSC San Diego) for sponsoring portions of this research. The author would also like to thank the following list of people whose support, assistance, and contributions made this thesis possible:

• Don Brutzman, from the MOVES Institute, Naval Postgraduate School (NPS) for being a pioneer, innovator and instigator of forward progress. His drive, dedication, intensity, and leadership are force multipliers for the Navy and Marine Corps.

Curt Blais for editing, encouragement, and vision. Curt provided an important sounding board for many of the ideas in this thesis.

• Curt Blais, Jeff Weekley, and Barb Helfer from the MOVES Institute, Naval Postgraduate School (NPS) for their guidance, support and positive reinforcement during the course of this thesis research.

• Eric Chaum, Naval Undersea Warfare Center (NUWC) for his encouragement and support in work with the Battlespace Generic Hub.

• Martin Reddy, Pixar Studios for doing a great job developing GeoVRML.

• Marion Ceruti, SSC San Diego for her advice and excellent papers on database technologies.

• Stephen Neushul, President and CEO, Radiographic Digital Imaging, for getting me started on data formats, and network distribution of large data files.

• Peter Neushul, Associate Professor, History of Technology, for giving me perspective on the state of technology and the capacity for institutional change.

• Andrew Neushul, Mechanical Engineer for leading me down the road of single board and embedded computers, which are essential for tactical systems.

• My wife Julie, son Jesse and daughter Jordan for their encouragement love and support. Jesse, age 5, provided valuable user interface testing and advice for the 3D globe and terrain navigation software.

• My fellow students (Claude (Odge) Hutton, Sean Hynes, James Harney, and the entire SAVAGE group) who challenged and inspired and inspired me with their accomplishments.

I. INTRODUCTION

A. OVERVIEW

This work represents the realization of Network-Centric goals of interoperability, cohesive battlespace visualization, information management, and systems integration. The application of structured data methodologies using the Extensible Markup Language (XML) allows organizations and systems to exchange and process battlespace information cooperatively. The application of this important technology is described and demonstrated.

Extensible Markup Language (XML) Technologies were developed to expand upon the success model of the World Wide Web. Applications that leverage XML are described, and tools that were developed to enable XML application development are presented.

Ongoing problems of interoperability, information overload, and battlespace visualization with military information technology systems must be addressed. The work is grounded in principles of leadership responsibility to control the data that populates information systems. The focus in this work is on the governance of systems by structured data, and the rejection of proprietary, application specific solutions that fail to meet to interoperability needs of the military. The role of leadership in this effort is defined as Data Control.

XML is used to address problems that represent a cross section of battlespace visualization and command and control (C2) applications. The representation of the physical battlespace is addressed in the delivery of 3D terrain views over a network. The tracking of friendly and enemy positions is addressed by the development of a standard, XML defined position reporting data structure. Information exchange and interoperability between joint and multilateral forces and between disparate systems and databases is addressed in the expression of an existing battlespace information exchange ontology using XML. The need to effectively employ autonomous systems in the battlespace is addressed using XML defined parameters that govern an artificial intelligence driven multi-agent community of unmanned autonomous aerial vehicles.

This work takes a pragmatic approach to implementing XML technologies, and offers a critical analysis of the problems associated with lack of interoperability between proprietary systems. A leadership driven, mission oriented approach is described.

B. THESIS ORGANIZATION

This work is exemplar oriented. Because the problem space is so wide, a cross section of basic command and control problems are addressed in order to demonstrate the efficacy of the ideas presented for different applications. The themes are extensibility, Network-Centricity, open standards, and Data Control. Each chapter addresses a basic concern for command and control, and presents an exemplar that illustrates proposed solutions.

Chapter III provides an overview of XML and the processes that can be applied using this important technology. The exemplar in this chapter is a Java code library that was developed during the development of the following exemplars. This code package, XMLTools, provides a beginning programmer with a basic toolset with which to leverage XML. The chapter describes key features of XML, and explains how software is used to implement them.

Chapter IV demonstrates the use of terrain data for battlespace visualization. Other important capabilities such as file management, and interest management for network accessibility of large datasets are demonstrated. The exemplar demonstrates the use of X3D to produce powerful web-based data access and visualization tools.

Chapter V addresses Geodesy, and the problems associated with establishing the geographic locations of enemy and friendly units. This has been an overriding concern since military operations became global in scope. The problem of position reporting is addressed in this chapter with the proposed use of a common XML defined language that position reporting applications might be required to use.

Chapter VI takes on the subject of battlespace information exchange and the conversion of existing databases for use in an XML driven Network-Centric environment. The focus of the exemplar is on the transformation of database schema to XML Schema, but the database that is transformed is one which has been designed to facilitate joint and multilateral interoperability.

Chapter VII introduces the implementation of autonomous agents in the battlespace. Multi-agent communities can be made up of different hardware systems can have varying capabilities, and must be controlled by warfighters in a way that produces optimum results. Interoperability between different systems must be addressed using standard data structures and command ontologies that apply specifically to the autonomous environment. XML-defined data control measures are illustrated.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. INTRODUCTION

This chapter describes the concept of battlespace visualization, and the principle of Data Control that must be applied to achieve effective information technology support for the warfighter. Principles of extensibility, and the roles of software are discussed. Current approaches are critiqued, emergent solutions are described, and the motivations for these efforts are explained.

B. PROBLEM SPACE

Battlespace visualization is a function of "Operational Design," a process by which commanders establish situational awareness, articulate the mission, isolate critical information, define centers of gravity, establish intent and direct courses of action.[1] Figure 1 illustrates the integral role that this process plays in the exercise of leadership in war.

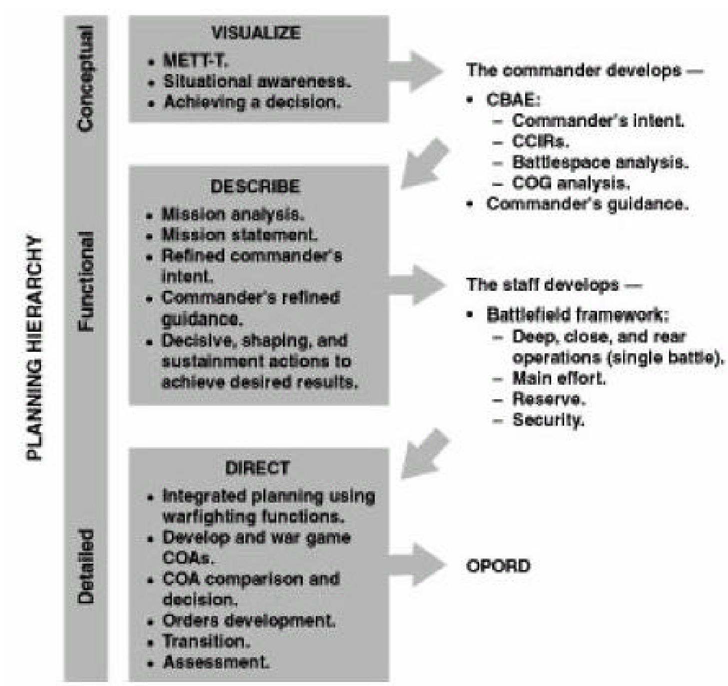


Figure 1. Operational Design[1]

Information technology is intended to assist and streamline the process of battlespace visualization, so that leadership can maintain control over a chaotic

1 Headquarters United States Marine Corps, Marine Corps Doctrinal Publication (MCDP) 1, "Marine Corps Operations", September 2001

and dynamic information-management environment. This is one dimension of "battlespace geometry," which is described as the "dynamic, multifaceted and multidimensional environment in which military operations occur."[2] The Global Information Grid (GIG),[3] the overarching architecture to meet this challenge, is described as a collective summation of information technology communication capabilities, and is the focus of network-centric strategies.

A Lessons Learned report from Operation Iraqi Freedom (OIF) describes an inability to effectively communicate battlespace geometry for planning due to software incompatibilities, and cited a lack of adequate National Imagery support during preparation or combat phases.[4] These observations describe problems with access to dynamic data as it is generated on the battlefield, and static data that has been collected and archived so that commanders can put battlespace information in a visual geographic context. There is a fundamental disconnect between the warfighter and the data that supports battlespace visualization.

Network-Centric Warfare is predicated on a requirement for "a strategic focus on interoperability."[5] The term "strategic" implies an overarching policy that permeates all levels of leadership. To achieve interoperability, and to ultimately enhance the human capacity for battlespace visualization in such a way as to promote intuitive information processing and decision making, this policy must be promulgated through operational courses of action, and the delineation of responsibilities. Within the systems of the military, the warfighters comprise the fundamental "software" that accomplish missions. Without active leadership, this software will fail. Likewise, it must be recognized that performance standards and interoperability of actual software in military information systems also require leadership supervision to succeed.

---

2 Headquarters United States Marine Corps, Marine Corps Reference Publication (MCRP) 5-12C, "Marine Corps Supplement to the DOD Dictionary of Military and Associated Terms", 1998

3 U.S. Department of Defense, *U.S. Department of Defense Directive (DODD) 8100.1: "Global Information Grid (GIG) Overarching Policy," The Pentagon, Washington, D.C.*, Sept 2002

4 United States Marine Corps, *1MARDIV Operation Iraqi Freedom Lessons Learned*, 2003, Topic: Battlespace Geometry/Zone Management, Topic: Lack of National Imagery Support

5 Department of Defense, *Network -Centric Warfare, Department of Defense Report to Congress*, 27 July 2001

A modeling process is required for computer representation of battlespace information, and a key task for leadership is to maintain control over the way that functional warfighting processes are modeled. Effective battlespace visualization is not solely the responsibility of hardware and software engineers, but rather must be under the full cognizance of military leadership. Organizations must inject their training, doctrine, and ethos into the data structures that they use to visualize the battlespace, so that principles of discipline and leadership are directly reflected in software tools. This is the principle of Data Control in the command environment.

To create software that can accommodate the vast and changing requirements of modern warfare, and that can leverage the rapidly evolving technological environment, it is important to focus on the human processes of battlespace visualization, and to identify those areas in which these processes can be enhanced. This is not just a problem of information management and graphical rendering. It is a complex and unending procedural problem space that must grow with, and respond to, the many and changing needs of the warfighter. The paradigm that best addresses this requirement in modern computing is "extensibility." This concept places a premium on the universality, adaptability, accessibility, and responsiveness of data, as exemplified by the most effective and far reaching interactive computing technology in history – the World Wide Web (WWW). The physical and cognitive connectivity that the WWW has demonstrated is the basis of Network-Centric Warfare.[6]

The "Cognitive Domain" is described in Network-Centric Warfare doctrine as "the domain where commander's intent, doctrine, tactics, techniques, and procedures reside (and that the) key attributes of the cognitive domain have remained relatively constant since Sun Tzu wrote *The Art of War*."[7] This is the realm of battlespace visualization that is represented at the "Conceptual" level in Figure 1. To the maximum extent possible, the expression and execution of the "key attributes of the cognitive domain" using computer technology must be

[6] Ibid. 5

[7] Ibid. 5

tightly governed by military leadership that is attuned to this arena, and cannot be relegated to the vagaries of "off-the-shelf" software. The Extensible Markup Language (XML)[8] provides key control measures that allow leadership to ensure that the vital process of battlespace visualization is not defined by software, but rather is expressed as an actionable standard to which software applications must conform.

Many of the concepts that are promoted in this thesis are distinctly contrary to existing architectures and processes. The pursuit of high standards for functionality and utility is a fundamental requirement for innovation. The approaches in this thesis favor open source software, non-proprietary data standards, and service-oriented development contracts. The emphasis is on total control of all data and software processes by military leadership. Chosen exemplars are intended to counter arguments that this level of responsibility is impossible to attain. Because there are significant economic and operational ramifications to the implementation of these methodologies, these are introduced as innovative disruptive technologies,[9] the consideration of which is a stated priority for the realization of Network-Centric warfare goals.[10]

C. OVERVIEW

1. Leadership

The first step in addressing the problems associated with data control and battlespace visualization is to understand the breadth of the problem space. This is an area that is as complex and varied as warfare itself. There are no "off-the shelf" software systems that will meet the needs of each and every commander on the battlefield. They must be given the means to meet their own needs. Once it becomes apparent that battlespace visualization is a data-centric endeavor that requires standardized data structures and control measures, then organizations will be able to take control of their data environments.

---

[8] World Wide Web Consortium, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation, October 2000

[9] Christensen, Clayton M., *The Innovators Dilemma, When New Technologies Cause Great Firms to Fail*, Harvard Business School Press, Boston Massachusetts, 1997

[10] Ibid. 5

An important cultural adjustment that must be made to incorporate data control and extensibility into the battlespace is to recognize that the responsibility for data control lies with leadership at all levels. Currently, it is acceptable to place the blame for interoperability and compatibility failures on “off-the-shelf” software that simply was not designed for the specific needs of the warfighter. A culture of data awareness and responsibility must be developed among leaders. If functionality is lacking, it must be addressed in the way that data structures are designed or in the way that they are presented in software applications. Active involvement in the constant adjustment of warfighting data by professional leadership is imperative for effective synchronization of human and machine processes. Just as mission oriented orders provide structure and guidance to military operations, so must data structures define interoperability and functionality for software.

A key factor in the development of computer software for battlespace visualization is the recognition of the primacy of the human mind in the role of leadership.[11] Decision aids cannot make decisions, visualization tools cannot make assumptions, analysis tools cannot guarantee ground truth, and detailed communication cannot not supercede battlefield presence. These things are the responsibility of the leader in war, and proper design for supporting this requirement must be explicitly acknowledged by software tools.

The prevalent human factor in computer aided battlespace visualization is “Information Overload.” Because there is so much data available, and because the effort required to filter and process this information is so great, it is difficult to apply the principles of instinct and experience-driven judgment to the contextual filtering of information. Commanders have little control over the data, and few ways of gauging dependability, accuracy and authenticity. There are currently few control measures that govern data formats, the location of data on a network, the categorization of data, or the encapsulation of data in semantically logical

---

[11] Headquarters United States Marine Corps, *Marine Corps Doctrinal Publication (MCDP) 1-0, "Warfighting"*, June 1997 ‘we do not believe in a formularistic approach to war—but in the mind of the Marine.”

data structures that can be effectively processed by software. Thus, not surprisingly, information overload is caused by immense amounts of unstructured data in uncontrolled data processing environments. These many challenges are commonplace, seldom resolved well, and yet are critical pre-requisites for success on the battlefield.

This thesis suggests that it is possible for military leadership to have the same role in guiding software and data functionality as it does in the maintenance of military organizational culture and operating procedures. Computer software can, and must, be custom fitted to accommodate organizational needs. This necessity must be acknowledged as a fundamental requirement before military software can become more of a help than a hindrance.

2. Application-Specific Software

Software that is developed as a product tends to create and maintain a self-contained data and presentation environment. It is usually operated by a limited community of professionals who are qualified to work with the product. Often it is possible to extend such an environment in order to provide greater functionality and for this reason many programs can claim to be extensible. Software that establishes a proprietary environment, and uses proprietary data formats maintains a high level of control over its data. By maintaining proprietary standards of extensibility and data control, software vendors can promise to accommodate the changing needs of customers as they arise, but then customers are denied the ability to accommodate their own long term needs.

By limiting open and independent extensibility, most often associated with open source software, vendors “lock-in” dependencies and are assured of future employment. Because they are reliant on proprietary systems, customers are assured of future expenses, limited extensibility, and no data control capability or responsibility. Commercial standards generally don’t require open extensibility, because that costs money, and can often be more work that it is worth to the immediate customers. Proprietary applications that allegedly provide full service data management and presentation are being challenged by open source software that, like the disruptive technologies described in Christensen, "often result in *worse* product performance, at least in the near term."[12] The goals of open source software place a premium on independence from the artificial limitations that vendors place on their proprietary solutions. Open source software is a positive disruptive technology that will allow Data Control by military leadership.

3. Data-Centric Software

Software that is designed to respond to structured data maintains extensibility and interoperability by conforming to common standards and reference models that are specified in data formats and structures. These standards and reference models can be applied to govern both content and visual rendering of data. A web browser is the most obvious example of software that is governed by structured data. Web browsers depend on instructions that conform to the specifications of the Hypertext Markup Language (HTML) and the Extensible Hypertext Markup Language (XHTML).

There is a great deal that can be done with software that is directly modeled after a web browser and functionality can be obtained by extending and enhancing current web browser technology. This is not, however, the extent of the potential that is inherent to data-centric software. XML is designed to leverage the principles that were proven so successful with HTML and to allow the creation of data languages to which software will respond. The XML technology that allows language development is XML Schema. Schema-aware software is designed around the principle that data is not simply content, but also includes "meta-data" that must be processed, interpreted, and obeyed. Data control can be achieved by the requirement that all software tools comply and respond to requirements set forth in XML Schemas. Proprietary software tools can meet these criterion as well, but they must relinquish the traditional crutches of exclusivity in data formats and presentation environments.

[12] Ibid. 9

4. Semantic Logic

Tim Berners-Lee, a leader in the development of the World Wide Web, has postulated that the internet of the future will be the vehicle for what he calls "The Semantic Web." This concept acknowledges the power of computer technology to apply techniques such as artificial intelligence, data mining, and autonomous agents in order to interpret, decipher and present information based on its semantic context, definitions, and organization.[13]

The significance of this capability is that it relies upon structured data that is designed specifically to provide context, meaning and organization that allows consistent and accurate data interpretation and delivery. This can either be a deliberate process, or can result from the natural accumulation of material that eventually is used to achieve a consensus on the appropriate meaning and consistent treatment of data as it is encountered in specific contexts.

The principal control measure that can be used to leverage semantic logic is ontology. This is a broad term that simply applies to meaning that is conveyed through language. XML Schemas that apply to specific data, and which encapsulate meaning, definitions, and usage parameters are powerful ontological expressions. To realize this powerful capability sooner, rather than later, it is important to begin the development of XML Schemas to establish a strong relative context for battlespace information.

As the new generation of semantically aware software is developed, battlespace information management and visualization will benefit from early, directed development. The creation of data structures that establish contextual meaning, assert appropriate translations, and define behaviors also encourages the exploration and analysis of current doctrine. More importantly, it allows organizations and services to assert and maintain cultural and mission prerogatives in the way that they implement information technologies. The Marine Corps, for example, has a vested interest in maintaining its unique and

[13] Berners-Lee, Tim, Hendler, James and Lassila, Ora, *The Semantic Web*, Scientific American, May 2001

powerful cultural ethos in the way that it is modeled by data structures on the GIG. For example, a request for close air support in a Marine Corps context will be interpreted differently in an Air Force context.

6. Validation

An often ignored requirement for truly extensible software is the capability for validation. Validation uses XML Schema to verify the structure if XML instances. The ability to validate the content of a file, text message, or binary packet against a specified Schema is one of the most powerful aspects of extensible software and of XML in particular. A goal of Data Control is to ensure that all information that populates the GIG can be recognized and validated in an application independent manner.

Validation is the tool by which Information Technology leadership can enforce control measures. An example of a potential application is the DoD Web Site Administration Guidance that stipulates required content and formats for all DoD web pages.[14] This effort is well meaning and well directed, but is impotent because it does not provide the means by which administrators can both develop compliant web pages and identify web pages that are not compliant. An XML Schema that takes advantage of the XHTML format, and stipulates specific design and content data structures, will provide the structure to accomplish both of these goals. XML Schema and XML validation can be used to control web content and design, as well as to ensure interoperability between systems' software.

7. Common Operating Environment (COE)

In order to affect change and take advantage of the extensibility paradigm in the military environment, it is important to recognize the existing infrastructure that governs all military Command, Control, Computers, Communication, Information, Intelligence and Interoperability(C^4I^3) software. The key to interoperability is the Defense Information Infrastructure Common Operating

[14] U.S. Department of Defense Office of the Assistant Secretary of Defense (Command, Control, Communications, Intelligence) , *Web Site Administration Policies and Procedures*, 2002

Environment (DII-COE).[15] Leadership must focus on this system of systems as an instrument through which the appropriate levels of software and data control can be obtained. This is, of course, an optimistic approach since the current implementations of the DII-COE are extremely platform and operating system dependent and have very few extensible characteristics.

An important positive aspect of the DII-COE is that it is supported by a well established and focused military-civilian culture that is dedicated to the establishment of common ontologies and data control mechanisms by which to achieve interoperability and cohesive battlespace representations. The XML based techniques like those discussed and demonstrated in this thesis can be applied within the DII-COE, in order to obtain full control over information structure and data exchange mechanisms that support the warfighter.

8. Joint Mapping Tool Kit (JMTK)

Another important focus for battlespace visualization is the Joint Mapping Toolkit. This is currently being implemented using commercial proprietary Geographic Information Systems (GIS) geodesy functionality in all C^4I^3 software. The requirements for open and extensible architectures are specified in the Functional Requirements Document (FRD)[16] for the Commercial Joint Mapping Toolkit (C/JMTK), but there is little in the proprietary software implementation that departs from the traditional "Off-The-Shelf" approach. The software is based on proprietary data standards, and relies implicitly on the Microsoft Windows® Common Object Model(COM)™ architecture[17].

Military leadership has no fundamental control over any of these proprietary architectures in the C/JMTK software and data that is to be used to support the geodesy requirements of the warfighter. They can be changed at the

[15] Carr, Francis H. and Hieb, Michael R., *M&S Interoperability within the DII COE: Building a Technical Requirements Specification*, Paper 00F-SIW-133 at the Spring Simulation Interoperability Workshop 2001, Orlando, Florida, March 2001

[16] National Imagery and Mapping Agency (NIMA), *Commercial Joint Mapping Toolkit (C/JMTK) Functional Requirements Document (FRD) (or C/JMTK FRD)*, 26 January 2001

[17] Environmental Systems Research Inc (ESRI), *Architectural Solution and Standards Compliance*

whim of the owning corporation, which can result in cascading upgrade and compatibility costs. This software may be vulnerable to attacks for which military leadership has no role in anticipation or prevention. Given the fundamentally vital role of geodesy data and software to all military operations this allocation of responsibility to private enterprise is irresponsible.

The appropriate requirements adjustment that must be made to this project in order to mitigate the situation is to stipulate that software not be reliant on proprietary architectures, and that all data formats must comply to standards and formats over which military leadership has complete control. This can be accomplished using XML Schema.

9. Innovation

Navy leadership has demonstrated a significant commitment to the principles of Network -Centric warfare, and has recognized the need to embrace innovation and change as constant forces in this arena. In order to take full advantage of the wealth of innovative potential in our warfighting communities it important to engage and empower duty experts and operators in the development of the structured data models that are to serve them. This is where the "human readability" principle of XML is very important.

Many baseline XML Schemas can be developed by going over existing orders and directives with duty experts and expressing them in hierarchically organized documents. If the basic premise of the exercise is understood – that of "writing orders for computers to understand," it is likely that individuals with well-developed professional knowledge will be able to derive innovative and useful expressions of the data that they use without having to become conversant in computer programming concepts. An important symptom of success for Network-Centric warfare will be a sense of involvement and engagement by military professionals from all functional areas as they begin the unending process of expressing their roles in terms that can be understood and reflected in extensible, Network-Centric software.

10. Disruptive Technologies

Success models are vital in the determination of requirements for modern military software. XML is a product of the World Wide Web Consortium (W3C) and is backed by the success model of the Hyper-Text Markup Language(HTML) that is the mainstay of the World Wide Web. XML supercedes HTML in the W3C with the advent of XHTML[18], and can now be found in all aspects of web communications. The significance of XML to the warfighter and to the task of battlespace visualization is in the size of the problem space that XML is designed to handle. Like HTML before it, XML can be made accessible to all comers. It is designed as a mechanism for defining languages, and as a format that anticipates and implicitly accommodates change. It is a tool for reconciling languages, and for establishing contextual logical relationships based on semantically defined parameters.[19] In the search for a control mechanism that will allow computer software to integrate with and enhance complex human processes and activities, XML is not to be ignored.

There is a clear bias against proprietary software models and proprietary data formats in this work. Although XML is extensively applied by many large venders, it is not always used to promulgate open standards or schema aware software. Profit is the dominant metric in commercial industry. The "Commercial-Off-The-Shelf"(COTS) approach to software procurement is highly influenced by profit motives. Data governed software and total customer control over data formats represent requirements that are disruptive because they challenge mainstream sensibilities that derive profit by denying Data Control.[20]

Network-Centric Warfare(NCW) doctrine states that "(NCW) is to warfare what e-business is to business." HTML was a disruptive technology in the way that it was used in the domain of e-business. Many companies that did not adapt

[18] World Wide Web Consortium, *XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)*, W3C Recommendation, October 2002

[19] Ibid. 13

[20] Ibid. 5

to the business models that the web promulgates met with failure. XML provides a similar movement in an emerging environment that is marked by connectivity. Those who cannot leverage connectivity by maintaining Data Control will fail.

XML technology is disruptive to the status quo of "Off-The-Shelf" software, and will redefine the concept of software development as a service vice product based endeavor. Software must be required to be as adaptable and flexible as the individuals that use it. Demands must be made that force extensibility and Network-Centricity. Current practices of software licensing at the expense of data control must be rejected, and solutions that enable common visualization, and that incorporate information exchange data models must be adapted at all levels.

11. Department of Defense Net-Centric Data Strategy

There is a tendency to downplay the role of XML in most discussions of Network-Centric strategies. In fact, the Department of Defense Net-Centric Data Strategy mentions XML only in passing as a small part of the DoD Metadata Registry.[21] The cursory treatment of XML in a strategy that is so heavily dependent on it reveals a reticent approach to a disruptive technology. The DoD is currently heavily invested in traditional relational database methodologies which will be transformed by XML technologies. The use of XML Schema to encapsulate relational database schemas, as demonstrated in Chapter V of this thesis demonstrates this potential.

In his Address to Joint Battle Management Command and Control Summit, Michael Wynne, Acting Under Secretary of Defense, described problems with interoperability that were defined 20 years ago after Operation Urgent Fury in Grenada, and that have remained unsolved through operations in Operation Iraqi Freedom. He asserted that the "a true Joint Battlespace management architecture.. is perhaps the single most vital warfighting

[21] U.S. Department of Defense, *Department of Defense Net-Centric Data Strategy*, May 9, 2003

technology for our military Transformation" and that we "need a Joint plug and play network that is self-organizing, and built using a mission-type, execution focused approach." [22]

12. Voluntary Consensus Standards

XML is a Voluntary Consensus Standard (VCS) that is used to develop languages that themselves are Voluntary Consensus Standards. The use of VCS in the government is intended to "Provide incentives and opportunities to establish standards that serve national needs"[23] The referenced circular expands upon the National Technology and Transfer Act of 1995 which seeks to " ..coordinate the use by Federal agencies of private sector standards, emphasizing where possible the use of standards developed by private, consensus organizations."[24] The intent is to create a convergence of government and DoD data formats with mainstream public data formats so that government agencies and warfighters can obtain access to all possible sources of mission critical information. In his article, "Marking Up Bureaucracy," Paul Ford describes the movement to require federal agencies to first use VCSs to "carry out policy objectives" and that "Increasingly, these VCSs are XML-based schemas."[25]

D. EMERGENT SOLUTIONS

1. On the Job Vice Off the Shelf

The potential of computer software to dynamically adapt to accommodate the unique needs of every user and situation has yet to be realized in the realm of "Off-the-Shelf" software. This potential is being addressed by web technologies such as web portals. These tools are XML enabled and represent favorable success models that are worthy of emulation.

[22] Address to Joint Battle Management Command and Control Summit by Michael Wynne, Acting Under Secretary of Defense, *Joint Battle Management Command and Control; Transforming the Battlespace*, July 30, 2003

[23] U.S. Office of Management and Budget, *CIRCULAR NO. A-119,Federal Participation in the Development and Use of Voluntary Consensus Standards and in Conformity Assessment Activities*, February 10, 1998

[24] United States 104th Congress, *Public Law 104-113, National Technology Transfer and Advancement Act of 1995*, 1995

[25] Ford, Paul, *Marking Up Bureaucracy*, Published on XML.com http://www.xml.com/pub/a/2003/09/24/government.html, O'Reilly and Associates, Inc., 2003

The distinction between "Government-Off-The-Shelf (GOTS)" software and COTS software is an indicator of a paradigm lag in both the military and commercial sectors. Software is currently considered as a commodity that comes complete with its own proprietary data formats, has a "cradle-to-grave" lifecycle, and is normally protected by licensing agreements that indemnify the developers against all levels of malfunction. This "money-up-front" business model is in direct conflict with the needs of the warfighter.

The extensibility paradigm takes the position that computer software is most effective when it is decoupled from the constraints of specific systems and platforms. This is a basic design tenet for XML enabled applications that is described as the separation of data and presentation.[26] Truly flexible and adaptable software is designed to respond to requirements that are defined using structured data. The most prevalent example of this kind of software is the web browser. This is a tool that is designed to respond to directives that conform to the HTML or XHTML data structures. The power and utility of this software is self evident. XML based software expands upon this model not by adding bells and whistles to the browser software, but by recognizing and interpreting powerful new XML defined languages that contain the requisite instructions for advanced functionality, with the same levels of customizability and flexibility that web pages exhibit. XML promises to extend the browser model to all levels of software functionality. Of course, it works in a web-browser as well.

2. Office Suites

Battlespace visualization takes many forms besides graphical representation. Tools include Operations Orders, Execution Matrices, and spreadsheets that chart the Order of Battle, Tables of Organization, and Time-Phased Force and Deployment Lists[27]. Unfortunately, these important devices are limited by the design prerogatives of a proprietary office format on a proprietary operating system. Even within these common operating

[26] Rosenthal, Arnon, Seligman, Len and Costello, Roger, *XML, Databases, and Interoperability*, Federal Database Colloquium, AFCEA, San Diego, 1999

[27] Headquarters United States Marine Corps, *Marine Corps Warfighting Publication (MCWP) 5-1, "Marine Corps Planning Process"*, January 2000

environments, incompatibilities abound that make the extension of these vital tools to a collaborative or data-fusion environment virtually impossible. The most irresponsible aspect of the use of these office products in the battlespace is that no military prerogatives have been incorporated into their design.

An important requirement for extensible battlespace visualization software is that all supporting processes be conducted using software over which full control can be leveraged by leadership. Duties that are performed using office type applications are prime candidates for data control. An important starting point for the development of office tools that are customized to the unique and specific needs of each warfighting functional environment is the OpenOffice.org[28] office suite. This is an open source product that produces native XML file formats. Leadership must recognize that "catch-all" Office suites are not suited to the highly specialized and critical functions that support planning and operations in war.

Because it is possible to provide open source custom solutions that reflect the unique vocabularies and requirements of each service and organization, this is where resources must be spent. Software must be customized centrally, starting with existing open source code, and must be distributed across the warfighting community. Currently the resources that would allow this progress are spent on individual licenses for software that is not designed for the specific purposes of warfare, and over which military leadership has no control.

3. Visualization of Physical Space

Maps and terrain are fundamental concerns for battlespace visualization and represent important confluences of requirements and solutions. Traditional methods for battlespace geodesy are focused on 2D cartographic products. These tools are invaluable to the warfighter because there are existing and well established map-driven operational procedures at all levels. The wall map is a staple of battlespace visualization that has yet to be significantly challenged by technology, and will remain useful. Current software products rely heavily on

[28] *OpenOffice.org, The Native XML Office Suite*, *http://www.OpenOffice.org*

direct digitization of these maps that results in fixed raster based data formats that are not conducive to on-the-fly representation of a dynamically changing battlespace.

Newer vector-based products use graphic objects instead of raster, or pixel specific data, to represent map information. This means that they can be updated efficiently in response to current reports, satellite imagery, and other inputs. Vector products can provide the most current 2D and 3D representations of the battlespace that are possible at any given time. Levels of detail can also be better represented because object dimensions can be adjusted to reflect perspective. An important XML based data ontology that addresses the requirement for this functionality is the Scalable Vector Graphics (SVG) language[29].

3D visualization capabilities directly address the challenge of providing intuitive advantages to the decision maker, and to enhancing the ability of leaders to process information quickly and effectively. This potential implies great responsibility with regard to accurate representation and to the avoidance of inappropriate visual representations that may convey artificialities that might result in bad decisions. An example of this is the overlay of 2D imagery on a 3D surface. If such a representation is produced without appropriate constraints, a user may be able to view the imagery from an angle that is not properly represented in the 2D image. The result might be an inaccurate impression of the location of certain key terrain features or obstacles due to artificialities caused by the adaptation of 2D imagery to 3D space. This is a consideration that drives home the need for absolute data control by leadership so that important visualization processes and requirements are not left to the whim of developers. It is possible to use XML to restrain 3D viewpoints, and to ensure that required perspective specific camera angle data is included with all 2D imagery.

[29] World Wide Web Consortium, *Scalable Vector Graphics (SVG) 1.0 Specification*, W3C Recommendation, September 2001

4. Extensible 3D (X3D)

An important XML based language that pertains specifically to battlespace visualization is the Extensible 3D Graphics Language.[30] The exemplar in Chapter IV demonstrates the principles of extensible, data controlled software with an X3D based application that allows web based delivery of navigable, battlespace aware 3D scenes. This software processes raw Digital Terrain Elevation Data (DTED)[31] and can be used for any number of Modeling & Simulation or C^4I^3 visualization purposes. It is operating system independent, platform independent, and is fully Network-Centric. X3D leverages the success model of web based technologies to achieve functionality that is not available in "Off-The-Shelf" software. X3D based software represents an important step toward web-deliverable interface functionality that is developed and maintained as a service vice as a licensed product. If data control and adaptability are priorities, Network-Centric, service-oriented software will become the norm rather than the exception.

5. The Battlespace Generic Hub

As important as the decision to implement XML technologies is the decision of where to apply them. Consensus standards are driven by existing requirements, standards, specifications, and ontologies, not by clever new data structures that claim authenticity only by the distinction of being written in XML. In the realm of battlespace visualization, a prominent and credible existing product is the Command and Control Information Exchange Data Model (C2IEDM) developed by the NATO Army Tactical Command and Control Information System (ATCCIS).[32] The C2IEDM is a well defined and established data model that is based on all of the common reporting mechanisms that are used across the spectrum of military operations. This data model can be expressed in several different XML Schemas. A definitive exemplar in Chapter V

---

[30] Web3D Consortium, *ISO/IEC 19775:200x, X3D, Information technology, Computer graphics and image processing, Extensible 3D (X3D)*, 2001

[31] Department of Defense, *MIL-PRF-89020B Digital Terrain Elevation Data (DTED)*, 2001

[32] NATO, Army Tactical Command and Control Information System (ATCCIS) Working Group, *The Land C2 Information Exchange Data Model, Working Paper 5-5, Edition 5.0, ATCCIS Baseline 2.0*, 18 March 2002

demonstrates a method by which an XML Schema is auto-generated from the written specification for the C2IEDM relational database. This project supports ongoing efforts to create an ontology for battlespace information exchange that can be used by all C^4I^3 and Modeling and Simulation (M&S) software. The result of this work is a functional Battlespace Information Exchange Markup Language (BIEML).

E. MOTIVATION

In the chapter, Preparing for War, of FMFM 1 *Warfighting*, the "two dangers (of) over reliance on technology" and of the "failure to make the most of technological capabilities"[33] are described. The current situation reflects failure on both counts in the reliance on proprietary software that is not interoperable, and the failure to leverage data control technologies to support the most basic military functions. FMFM 1 also states that when technological dependence becomes a problem, "doctrinal and tactical solutions to combat deficiencies must also be sought." This work seeks to introduce the Data Control principle as a doctrinal approach to Information Technology implementation, and to prove its efficacy using exemplars.

The basic battlespace visualization functions that are addressed in this work include position reporting, terrain representation, battlespace information exchange, and autonomous aircraft control. Each of these subjects merits far more attention than the scope of this treatment allows, and the exemplars are meant to be taken as suggested starting points for the full development of extensible, interoperable solutions, or for the integration of extensible methodologies into existing software tools. Also addressed is the need to articulate requirements through the software acquisition system.

1. Terrain Visualization

Use of terrain elevations for battlespace visualization and decision support is not prevalent in current operations. Although the "lay of the land' is a principle concern of the infantryman, few tools exists that allow warfighters to leverage the

[33] Headquarters United States Marine Corps, *Marine Corps Doctrinal Publication (MCDP) 1-0, "Warfighting"*, June 1997

considerable collections of terrain elevation data that exist. The Exemplar in Chapter IV introduces a new capability for battlespace visualization, and demonstrates a way in which this capability can be made available to every level of command using simple, widely available web browser software. This capability represents rapid access to “fly-throughs,” and allows a squad leader to “walk-the ground” prior to a mission. The ability to produce 3D views on computers and handhelds is a product of rapid advances in processor and display technology, and of XML based Data Control measures.

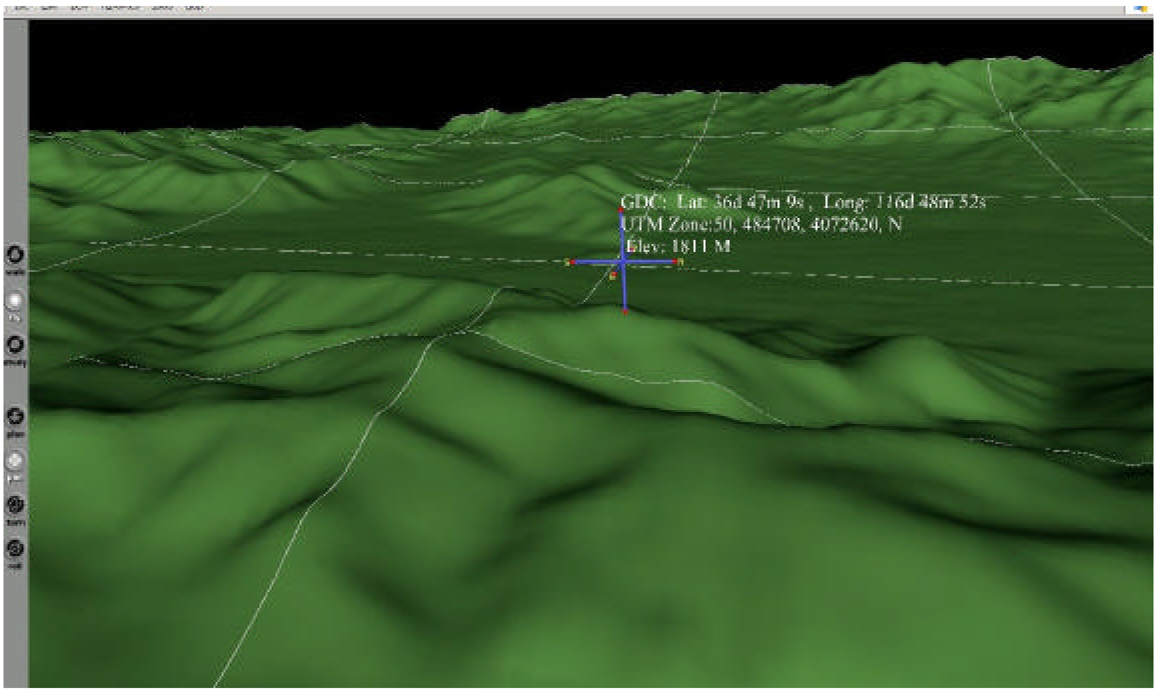


Figure 2. X3D Terrain View

2. Position Reporting

Problems with position reporting capabilities were evident in several “After Action” and “Lessons Learned” reports from OIF. The First Marine Division’s Lessons Learned described the Blue Force Tracker (BFT) and the Mobile Data Communication Terminal (MDACT) as incompatible systems, and ascribed the problem to communications differences.[34] This is a typical assessment on a level that does not recognize the role of software in these tools. Although each system has its merits in specific environments, neither of them is capable of

[34] Ibid. 4

producing a standardized, common messaging and position report format that can be interpreted universally by all command and control software. The BFT system was the preferred device, because of its range, but it could not talk to units using the MDACT[35], and could not be used to update the standard Marine Corp battlespace visualization software with position data.

This work does not seek to argue the merits of one system over another, but rather to promote the injection of data control into the vital process of position reporting. Vehicle, units, and individuals will need to use several different systems with which to report their positions, either manually or automatically. An imperative for all of these systems is that they provide receiving software systems with a standard known format, designed and mandated by leadership on the DoD level, so that efficient visualization can be accomplish using C2 software. The exemplar in chapter IV suggests a simple XML based format that can be used as a starting point for this important Data Control measure.

If Blue Force Tracker does become a widely used system, it must be mandated that it produce XML defined data formats that comply to DoD standards and requirements. Position reporting is one part of the larger problem of information overload, system incompatibility, and lack of data control that can be mitigated with Data Control.

3. Battlespace Information Exchange

Perhaps one of the most ambitious and far-reaching efforts in this work is the automated creation of an XML Schema to represent a prominent information exchange database mechanism. Virtually all interoperability problems on the joint and multinational levels are addressed by such a mechanism. An important step toward being able to require that software process data in an interoperable fashion is to create robust schemas that can accommodate the vast information exchange requirements of the modern warfighter.

---

35 United States Marine Corps, *Marine Corps Systems Command Liaison Team, Field Report, Central Iraq*, 20 April to 25 April 2003

4. Unmanned Aircraft Control

UAV technology is the subject of positive remarks in OIF after action reports, and is a key area in which technology must be fully leveraged in order to maximize its warfighting potential. An important tool in the process of battlespace visualization, UAV technology, also presents complex command and control problems, which are associated with organizational and logistic issues that must be addressed to employ this technology in an optimum fashion.

Interoperability will also be an issue as different platforms are employed by joint and multinational forces. Again, the establishment of standard XML defined message formats that dictate content and behavior for automated equipment is a requirement for interoperability. This project also addresses the application of agent based behaviors to facilitate control of these devices.

5. Software Acquisition and Development Principles

Software must accommodate the needs of the Warfighting community for which it is maintained. It is unconscionable to permit the adjustment of custom or doctrine in order to accommodate software, when it is far more advantageous to demand the opposite. Because leadership is traditionally focused on people, there is a strong tendency to accommodate bad software through training and education. The challenge to leaders is to demand software interfaces that require no training other than that which is inherent to the professional military activity. This of course is impossible without the ability to adjust and customize software interfaces. The extensibility principle of the separation of data and presentation places this capability on the difficulty level of web page development. Software must be trained, not the users.

It is all very well to postulate grand schemes for attaining Data Control and software that adapts to every need and context, but it is important to recognize the importance of the acquisition requirements process in the military environment. If the principles of Data Control and extensible software are recognized as important tools then this must be expressed in requirements for all software that is currently being developed, and that will be developed in the future. An important objective is a set of blanket requirements that stipulate that all software will communicate in accordance with organizationally developed XML Schemas, and that no proprietary data formats will be permitted.

F. CONCLUSIONS

Information technologies have great potential for enabling interoperability, battlespace visualization, and Data Control. The ideas in this chapter describe a pragmatic approach to realizing this potential by the assertion of fundamental leadership responsibilities. The measures proposed require cultural and organizational recognition of the problems, rejection of failure models, and a resolved and unwavering commitment to total ownership of information.

Sweeping proposals for change are ineffective unless direct action is specified, mission oriented procedures are implemented, and leadership supervision is applied. This work constitutes a challenge to all levels of command to take control of battlespace data.

THIS PAGE INTENTIONALLY LEFT BLANK

III. XML APPLICATION DEVELOPMENT

A. INTRODUCTION

This chapter describes the fundamental principles of XML application development. XML Schema and XML Transformation, are explained. Software that was developed to support the exemplars in this thesis is described, and a hypothetical example that addresses establishment of a validatable Network-Centric information domain is presented.

B. OVERVIEW

1. What is an XML Application?

An XML application is a combination of XML documents that are leveraged by generic software tools which are compliant with standards set forth in the W3C XML Specifications. XML applications are used throughout this work to demonstrate the potential and validity of this W3C success model. In order to understand the paradigm that XML promotes, an understanding of its key features is necessary.

2. Information Management and Databases

Information management requires information control. Lack of control prevents the realization of important capabilities. Nowhere are these capabilities more necessary than on the modern battlefield. There is an assumption, for example, that any data that exists in digital form on a network can be made available to any users or applications on that network. The reality, of course, is that organizational and logistic limitations prevent the effective distribution of information. Information overload, and inadequate database functionality are significant problems. Even though a document might be available, how is it to be located? How can users access only those parts of it that they are interested in? How can data participate in a larger information analysis scope in which only specific details are used? These are issues that need to be addressed explicitly if information systems are to reach their potential for battlefield support.

A common approach to solving information management issues is the implementation of databases. When data is committed to a database it is manipulated and controlled by a mechanism called a database schema. This is usually represented using a set of tables and diagrams that describe all of the entities and relationships that the database maintains in order to allow storage and retrieval of information in an organized and consistent manner. Often the database schema is graphically represented by something called an Entity-Relationship Diagram.[36] The conceptual design of a database will determine the means by which access, distribution, filtering, and analysis of data is to be accomplished. Data in a database is locally controlled, but outside of the database it is still subject to problems of accessibility and distribution. The database as a whole is subject to the same organizational problems of location, access, and scoping that characterize raw data.

The approach to organizing databases has been described as a "System of Systems." Basically this involves the creation of common lines of communication between typically very large databases. As the databases grow, and the lines of communication increase, the governing schemas become more and more complex. Eventually control is lost due to the sheer complexity of the arrangement. Even when it is functional, there is always a great deal of information outside of the "System of Systems" that is not under control, and yet requires management. This data is not subject to the established database schemas, and thus cannot be integrated into the systems. Unfortunately, most information that is generated using the common office suite software for planning and operations is uncontrolled by traditional databases. Very little is being done to populate database systems with traditional plans, orders and directives for selective access and reference in support of operations.

3. Common Data Formats

The reason that much information is not controlled by information management systems is because common data formats are not currently implemented. Office-suite document formats vary over time, and require specific applications for access. Data formats that are consistent over time are a

[36] Chen, Peter Pin-Shan Chen, *The Entity Relationship Model, Toward a Unified View of Data*, ACM Transactions on Database Systems, pp. 9 - 36, March 1976

requirement for responsible data control. The advent of the HTML driven Internet illustrated the power that a common data format lends to data organization. The basic set of rules that govern HTML are used to author, distribute and retrieve vast amounts of data among a hugely diverse population of users. The example of the Internet represents the existing potential for battlespace support. Realization of this potential has been described as "Network-Centric Warfare," and is considered to be a primary goal in the DoD[37]. The logical descendant of the principles that HTML represents is XML, and is a key technology for data control and Network-Centric information management.

C. XML SCHEMA

1. What is XML Schema?

XML represents the leveraging of the common format concept in a way that takes advantage of database techniques for the logical organization of information. Instead of one common set of rules, XML is a set of rules with which an infinite number of rule-sets can be defined. These rule sets are defined using a mechanism called XML Schema[38], and contain most of the functionality of traditional database schemas. The rule sets themselves are valid XML documents, which further reinforce the ability to maintain consistency. There is an XML Schema for XML Schemas. A traditional limitation of common formats has been the fact that no single language can accomplish all goals. XML addresses this limitation by providing the ability to create languages, along with the ability to provide explicit descriptions of these languages so that they can be effectively translated into any other XML based language. The plurality that XML brings to the concept of common formats makes it one of the most significant technologies in existence for data control, database interoperability, and information access.

The use of XML Schema to regulate adaptive, expanding, and extensible Network-Centric information systems is conducive to Data Control because the process of data definition and data generation are linked. Data is generated in

[37] Ibid. 5

[38] World Wide Web Consortium, *XML Schema: Formal Description*, W3C Recommendation, March 2001

accordance to XML Schema that can be published and used by target applications. As schemas are developed to represent the myriad different data systems, central schemas can be developed in order to create many-to-one focal points for data conversion. Standards can be implemented by the publication of XML Schemas. Local common languages can be established, and maintained within organizations, while outputs can be translated to global common languages. If the traditional "System of Systems" can be expressed as a "System of Languages", then XML can be used to bring virtually any kind of data under control.

XML Schema provides a valuable, system independent methodology with which information control can be obtained and maintained. If all data adheres to known schemas, then it can be integrated into a common system. Of course the extent to which the data can be integrated depends upon the astuteness of the XML Schema document designer. For this reason, the focus of information control and information management must be on the development of XML Schemas that define information resources and that govern all information generation points effectively. Information can be considered independently of containing databases or files, as long as its governing schema is known, and as long as it can participate in a global information distribution, collection and analysis system.

2. XML Schema Design and Development

Just as the organizational capacities of human beings are limited, so are the ways in which data can be organized. To a large degree, most data has already been placed in a format from which an XML Schema can be derived. There are many opinions on the way that XML Schemas might be designed, with many taking the view that data must be represented in such a way as to maximize the effectiveness of the current XML technologies. Often this involves complete redesign of existing databases, and as such is effectively unrealistic and unnecessary. Databases don't have to be re-resigned to conform to a new standard, only the data inputs and outputs need to be transformed. The most important factors in the extension of information systems using XML are data ownership and information Control. Information control cannot be achieved without the explicit participation of data owners, and a certain degree of inefficiency must be accommodated in order to maintain this critical link.

Because XML is extensible, XML Schemas can and will be changed. The extent of change will of course impact implementation. Since change is a constant, it is the responsibility of systems designers to incorporate the ability to accommodate change into their applications. The predominant role of data and data definition over system design makes XML technologies data-centric vice application-centric. Instead of using database applications to manipulate and control information, explicit data description is used to control the applications. An example of an application that relies on data to define its functionality is the standard web browser that responds to data in an HTML or XHTML format.

3. XML Schema and Structured Data

Information that is contained in an office style document is compliant to a format, be it XML or not, that governs presentation. User applied constructs that impose structure on data exist in the form of standardized document formats such as the Naval Letter Format, the 5 Paragraph Order, and the Operations Order Format. Within these documents and in attachments and appendices, data is often placed in tables in order to further encapsulate concepts. The fact that these formats are reasonably predictable and consistent means that they can be processed automatically. Currently little attention is paid to machine readability for documents that support mission requirements. As the benefits of a structured data approach become apparent, tools will be developed to assist in the production of data that is both human readable and machine readable. Developing XML Schemas that define the structure of current document formats must be a priority in this effort. The data transformation example in Chapter V demonstrates how structural and logical data from standard text formats can be used to auto generate a functional XML Schema.

4. XML Schema and Validation

An important aspect of standardization and data control is enforceability. While recommendations and requirements can be levied quite easily, it is another matter to ensure that all data is compliant. Validation is an important aspect of XML that has been included as a central design concept. XML Schemas are used to validate instances to ensure that the parameters and structure are compliant. Validation is an important requirement for Data Control because it ensures that all data follows parameters set forth in an established and publishable XML Schema so that software can adapt to different and changing data formats. XML documents can specify their governing schemas and thus contain all the information necessary for processing and presentation. This allows network delivery of data between applications and databases in a way that accommodates change and allows constant validation of data structures and content. Without validation, data control is notional at best, and information management stops at the local desktop.

5. XML Schema and Leadership

The responsibility for the design, control and maintenance of XML Schemas falls upon the using communities, and can not be relegated exclusively to programmers. To a large degree these communities already have the basis for XML Schema definitions in the standards, directives, specifications, and orders that govern the way they do business and the way that they generate and use information. In order to maintain a strong link between these governance tools and information management systems it is imperative that a direct correlation is maintained between directives and data. Organizational changes and improvements must be reflected in the XML Schemas in accordance with changes in the governing documents. The most important capability that XML methodologies bring to the military environment is the ability to create direct links between command leadership and information systems.

Once leadership decides that it will use the current methodologies of orders and directives to govern both human processes and information management processes it will become apparent that very little needs to be added. It will also be evident that current methodologies for expressing textual information must be subject to information control requirements if they are to be used in any data-centric capacity. In short, all orders and directives that impact information generation must themselves be rendered in XML. The information that is contained in these directives can then be directly and actively transformed and expressed as XML Schema in such a way that source document changes will be automatically reflected by subordinate information systems.

The direct alignment of XML Schemas with orders and directives represents the critical information control link that is required by the unique Network-Centric requirements of the military. Systems of computer systems and systems of human systems are all governed by explicit orders and directives. Using these tools for information control combines logical functionality and organizational authority to accomplish the task of information management.

If leadership can have direct control over information systems by the way in which governing documents are expressed, then Data Control is possible. Documents will use XML techniques to designate important concepts in a concise and well delineated fashion. It is important that XML Schemas for military applications are based as directly as possible on current directives. To accomplish this, a process of XML conversion and interpretation is necessary.

D. XML TRANSFORMATION

1. What is XML Transformation?

The advantage of using a reliable, standards based data format is that platform independent, dependable, consistent, and generic APIs can be developed to manipulate and control information. Reliance on stovepipe database systems and proprietary versions of the Structured Query Language (SQL) have established barriers between existing databases that prevent extensibility, interoperability and data control. The primary requirement for an XML data structure is that it can be reliably transformed and adapted for storage or presentation in an application and platform independent manner. XML Transformation is required for information control.

The most common use of XML transformation is in the presentation of data. XML formatted data can be manipulated using the Extensible Stylesheet

Language for Transformation(XSLT)[39] or byte code to provide different views for the same data. Usually this involves a transformation from XML to an HTML, or XHTML, format. The flexibility of this web based presentation method allows the same data to be made available in different ways, depending on user defined parameters. A mobile user will receive a compact version, a desktop user will receive a large page, and a wireless user will receive a smaller download. The output may also respond to inputs in order to create a customized information view. This is the basis of Web Portal technology[40]. All of this functionality is available by separating data from presentation using the principles of XML.

2. Database and Application Interoperability

A less visible, and yet equally powerful function of XML transformation is in the creation of conduits between databases and applications that can be used to establish control over information. Although information control and manipulation requires data format control, applications are necessary to maintain data structures and expose functionality. Many of these applications currently exist, and are extremely powerful and effective in processing the data formats that they were designed to support. The functionality of these applications can be retained in a context where inputs and outputs are controlled by XML transformations. These transformations ensure that data outside of the applications is logically controlled. An application or database can be extended to participate in an XML environment without redesign or alteration of internal data formats and processes.

Transformation can be performed using XSLT or with established APIs. An XML transformation can be designed to accommodate a specific XML Schema, so that it can consistently operate on documents that are conformant. XML Schemas that are extremely generic can produce an infinite number of instances. This is the case with the Schemas that govern text and office type

[39] World Wide Web Consortium, *XSL Transformations (XSLT)*, W3C Recommendation, November 1999

[40] Commonly referred to as simply a *portal*, a Web site or service that offers a broad array of resources and services, such as e-mail, forums, search engines, and on-line shopping malls. *Webopaedia.com*, *http://www.pcwebopaedia.com*

documents. These documents can be transformed on the presentation level very consistently. On the data level, however, it is necessary to impose logical considerations to apply transformation templates that rely on the data that is in the XML documents. Because this data can be infinitely varied, it is necessary to establish more strict control over documents that contain functional data.

The design of the XSLT language accommodates both generic and specific cases by using a template based process that applies transformations to segments of data using prioritization of specificity. This means that a standard, or default transformation will occur when a more specific treatment is not specified. This methodology is extremely powerful in the accommodation of change in data formats because it is difference based. When two Schemas represent similar data, but have a few critical differences in format and data relationships, an XSLT script can be written to address only the differences. Data that does not need to be transformed is passed through transparently. As change occurs, XSLT procedures, called templates, can be added to accommodate only these changes without having to redesign the entire transformation.

3. XML Representation of Databases

One of the disadvantages of traditional, table based databases is that they must be described in a way that is separate from the way that they are instantiated. A traditional database schema uses diagrams, text and tables to illustrate entities and relationships in a purely human readable format. This format is not intended to be machine readable so there is no automatic connection between the human readable schema description and the machine readable format which is usually expressed using SQL. Because there are various different versions of proprietary SQL, a database schema can be instantiated in incompatible formats. For this reason a requirement to comply to a schema is not sufficient to ensure data control.

The description of database relationships using XML Schema is fundamentally different in that the data structure is tree based vice table based. The hierarchical relationship information that is conveyed by parent/child and sibling configurations is often more intuitive than the key based system that table based structures employ. XML Schemas that represent table based databases can include key information in order to facilitate two way communication between table formatted data and tree formatted data. An XML Schema can be annotated extensively in order to describe the various entities and relationships. An XML Schema can be multi purposed to create database instances, to generate human readable reference documentation that describes the database structure in detail, and to validate instances. An XML Schema is an XML document, and can be validated in accordance with the W3C standard. XML Schema was designed to be an efficient encapsulation of data structure information that can be used as a basis for instance generation, validation, and transformation.

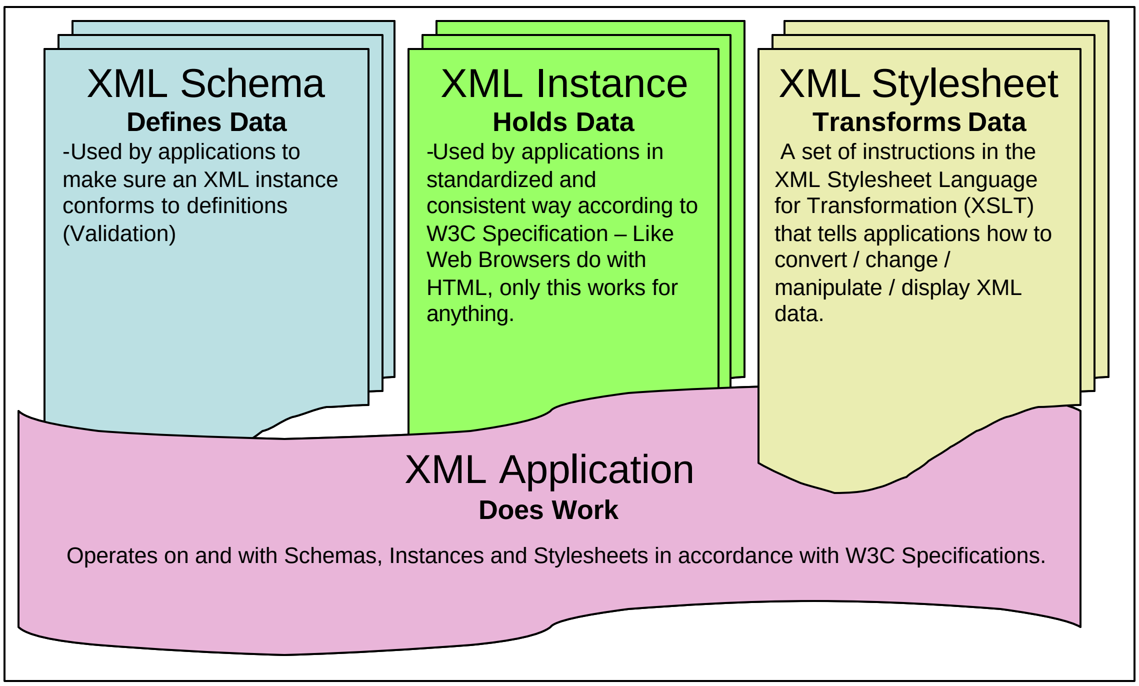


Figure 3. XML Application Anatomy

E. EXEMPLAR: XML TOOLS

1. What is Required to Use XML in Software?

XML is for all intents and purposes, inert. It doesn't do anything other than properly structure information. XML functionality relies on tools, just as HTML functionality relies on a web browser. The difference is that XML tools take the form of complete Application Programming Interfaces (APIs), and so incorporate the entire spectrum of computer data content manipulation and presentation. XML tools can be designed to handle specific XML documents, or to accommodate all XML in a generic fashion.

To a large degree, the W3C XML Specification is a set of rules that govern the way that software will treat XML defined information. The core applications that give XML relevance and comply to the W3C specifications are XML parsers and XSLT engines. These tools can be extended using standard APIs for C++, Java, and others.

The following sections describe Java programs that leverage open standards based, open source software APIs to achieve the basic functions that are required to use XML in software. Some tools are not XML specific, but perform convenience functions that are often needed in this environment. All of these classes are static, so they do not have to be instantiated by using programs.

2. JDOM

JDOM is an Open Source API that allows intuitive Java centric parsing and manipulation of XML documents. The use of Java to manipulate XML documents is far less extensible than the use of XSLT, but is sometimes expedient. An important aspect of XML application development is the minimization of byte code, in favor of the more accessible, and extensible XSLT mechanism. Most of the XML tools utilize the JDOM API.

3. Apache XALAN

XSLT Transformation engines are built into all web browsers. In order to produce an application that performs transformations without implementing browser functions, it is necessary to use a standalone XSLT engine. XALAN is an open source, standards based product that offers functionality without the need for proprietary software. XALAN is included with the Sun Java distribution, and so is commonly available.

4. XML Tools

The following Java code was developed to implement XML technologies in the conduct of this thesis. The subsequent exemplars use these utility classes.

They use open-source libraries, such as JDOM and XALAN described above and are hereby offered for free and open use. The physical publication of this thesis includes a CD Rom of all source code, and instruction to download these resources are provided in Appendix A.

a) Archiver.java

XML is sometimes criticized for being too verbose, and for creating files that are too large for web delivery. However, because textual markup is very repetitious, XML compresses extremely well. OpenOffice,org native XML format files are stored in archived files which typically makes them much smaller than their MSOffice counterparts. It is useful to have a compression/deflation capability when working with XML in software. Archiver.java can be used to archive a directory that has been mapped using XMLDirectoryMap.java. This allows automatic implementation.

b) BitReader.java

The ability to read Binary data is very useful. All binary formats can be mapped using XML Schema, and these mappings can be used to read the files. This utility is used by the DTED reader application to selectively read terrain data from within very large binary files.

c) ErrorDialog.java

This is a simple window that sends a message to a user. This is included as a useful mechanism for troubleshooting, as well as a tool that can be used to display the results of validation checks on XML documents.

d) GetInetInfo.java

XML is very useful for maintaining user and state data for web applications. This utility obtains that data.

e) IEBrowser.java

It is often useful to launch a browser to view XML documents, or the results of XML transformations. This performs that from a running process.

f) JDOMConvert.java

JDOM document objects use a different Document Object Model(DOM) than the one defined by the W3C to manipulate an XML document in byte code.

This utility converts to the W3C DOM so code that does not use JDOM can use the data objects.

g) JavaConfig.java

This is a simple program that creates an XML document to record program location and information in a user's home directory. This can be implemented by any Java program and simplifies installation. If software is to be automatically distributed and updated over the network, a mechanism like this can be used to track software and workstation metrics.

h) LoadXMLDoc.java

This utility simply reads in an XML document so that it can be accessed and manipulated using JDOM.

i) MakeDirectory.java

This simple utility creates a directory on a computer. In combination with XMLDirectoryMap.java and an XML Schema that defines a required directory structure, this can be used to automatically create directories which can then be maintained and validated over the network. Directory management is a key requirement for Data Control and is necessary for almost all software development problems.

j) NetAccessDialog.java

Often security must be implemented to access information on a network. This is a form that allows a user to enter a login and password.

k) StylesheetCache.java

This is a very powerful tool by Erik Burke, that allows a program to store in memory all of the XSLT stylesheets that it uses[41]. This saves time by not having to load the stylesheets repeatedly from disk. This is used by XSLTransformation.java.

l) VRMLMaker.java

A tool for applying the stylesheet that is used to transform X3D into VRML script for display in web browser plug-ins. This is an example of non-generic code that has been superceded by XSLTransformation.java.

[41] XSLT Processing with Java, Related Reading, Java and XSLT By Eric M. Burke, Published on The O'Reilly Network (http://www.oreillynet.com/)

m) WriteXMLDoc.java

A simple class that writes an XML document from JDOM to file.

n) XMLDataHandler.java

Performs some basic manipulation of XML Elements and Attributes using JDOM. These functions are performed much more handily using XSLT.

o) XMLDirectoryMap.java

When passed a root directory location, this class creates an XML document that represents the directory. The XML Schema that it follows in the production of the XML mapping is SystemDirectory.xsd. This is a very basic schema that will handle any combination of directories and files. This is a starting point for development of an XML Schema to dictate required directory structures, directory names, and file names so that they can be validated and accessed from a Network-Centric enterprise system.

p) XMLLegalize.java

When source documentation is used to auto-generate XML Schemas it is often necessary to create XML Element names and Attributes. This utility alters any text so that it will conform to specified XML formatting for Element and Attribute naming.

q) XSLTransformation.java

This is a powerful class that is the result of a cumulative learning experience in the development of XML based applications that use XSLT transformations to accomplish difficult tasks without having to create compiled byte code. This code allows consecutive XSLT Transformations that apply the output of one transformation to the input of another. Parameters can be submitted for XSLT execution and the StylesheetCache.java utility is implemented to store stylesheets in memory. This class is a key exemplar that demonstrates the power and flexibility of software that is driven by XML and XSLT. It uses the Apache XALAN transformation engine for Java.

r) XSLTransformTool.java

This is a basic Graphical User Interface (GUI) that applies XSLTransformation.java using entered XML and XSLT documents. This functionality is available in most XML authoring tools, and is integrated into web browsers, but this demonstrates the use of the capability in an independent application.

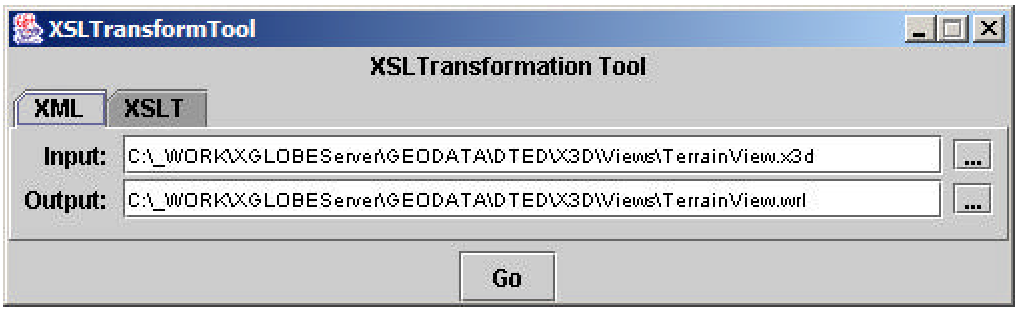


Figure 4. Screenshot of XSLT Transform Utility

F. A ROADMAP TO NETWORK-CENTRIC DATA ACCESS

1. Where's the Data? An "I Have a Hammer" Approach

A common epithet remarks that "When all you have is a hammer, everything looks like a nail." XML enthusiasts are often accused of this mentality. This is sometimes appropriate criticism, but often it reveals a misunderstanding of the true power that XML represents as a Data Control mechanism. This section poses a basic, practical implementation question related to achieving, or at least starting down the path of, the Network-Centric paradigm.

2. Establish the Information Domain

Network-Centric Warfare doctrine stipulates that "The force (must have) the capability to collect, share, access, and protect information."[42] Presumably this must somehow be implemented in a useful way. Currently, the capability exists but it is used very little. Although computers are interconnected they are fully reliant on a proprietary operating system to implement file sharing and data access. All data sharing is done using email or operating system dependent sharing mechanisms which offer little or no organizational control, and cannot be automated independently of the operating system.

A basic requirement to establish an information domain that truly allows data collection, sharing, access and protection is the implementation of an

42 Ibid. 5

application independent system for specifying the location of data on computer systems. Once this can be specified and enforced, then data location can be published to the enterprise so that it can be shared, accessed, and protected. Until a methodology is established that governs where data is stored on computers, Network-Centric warfare will remain a "future capability." Once the methodology and tools for controlling data locations are standardized, specified, and published, then applications will be able to perform required data access and protection tasks.

Why must this system be application independent? Currently this capability depends on the operating system application, which introduces incompatibilities and complexities over which leadership has no control. Important unit reporting procedures depend on reliable methods of file publishing and discovery. Leadership cannot relegate that responsibility to a single application, but must rather establish a methodology to which all applications must subscribe in order to access information. This can be accomplished using XML Schema.

Step 1: Top-Down Leadership: Develop a baseline XML Schema at the highest level that dictates directory structure for all subordinate units. This might contain directories for Reports, Plans, and Communications. Subdirectories might apply to Personnel, Equipment, Orders, Messages, and Organization. Of course, this is not a task to be taken lightly, but it is one for which leadership must take responsibility. It must also be very basic and simple to allow for straightforward implementation and extension down the chain.

Step 2: Bottom-Up Execution: Subordinate units extend the baseline Schema to accommodate specific needs. The structure and file naming formats dictated from next higher headquarters must be adhered to, but additional directories and files can be added to the subordinate XML Schema. Again this is the organizational responsibility of leadership. This process must be repeated to the lowest computer-using element in an organization. In the Marine Corps, this might be the level of a Platoon Commander. All Schemas are published in a specified location that is designated in the baseline XML Schema.

Step 3: Implementation and Validation: Simple utility programs – made available by higher headquarters on the intranet – are used to create the directories and name files in accordance with the XML Schemas developed by leadership. Periodically all computers are scanned using something like the XMLDirectoryMap.java utility and XML documents are created that map the directories. These documents are validated with respect to the established Schemas and discrepancies are corrected by leadership.

Step 4: Adapt and Improve: Schemas will require adjustment and improvement over time. As this occurs, utility programs can use XSLT to update subordinate schemas and apply basic directory management processes to adjust directories and rename files. Because strict Data Control is established and maintained from the outset, adjustment is possible. All software must use the published XML Schema to access and publish information on the enterprise.

Step 5: Protect: Security is a concern. This Data Control methodology implements "common sense" control measures that enhance security. When the specific location of data is known, access by humans and applications can be monitored, security measures can be applied at the file and directory level, and intrusion can be discerned quickly. Efficient redundancy and backup measures can be implemented. Current security methods attempt blanket measures to guard entire systems and computers. This is unwieldy, inefficient, difficult to track, and also impedes authorized data sharing operations. The organizational methods suggested here will greatly enhance security, and more importantly will shift the responsibility for security away from vendors, and toward leadership where it belongs.

3. Is This a Nail?

There are plenty of solutions to the problem addressed here. Operating systems are meant to solve it. Web services might attempt it. Certainly overarching database programs will promise to accommodate the need for basic file organization. These solutions require massive investments in proprietary systems, and yet still rely on procedures over which leadership has no control. More importantly, these tools do not yet exist in a feasible, extensible, open standard form. For lack of a better tool, this is a nail for the XML hammer.

This approach does not address the formats of the files in the directories, but rather only where they must be located and what they must be called. Eventually it is assumed that most of these files will be in an XML format so that they can become part of a distributed database like the one that is commonly accessed on the WWW using tools such as Google[43]. This methodology will lead to the development of an operational battlespace search engine that already knows where everything is.

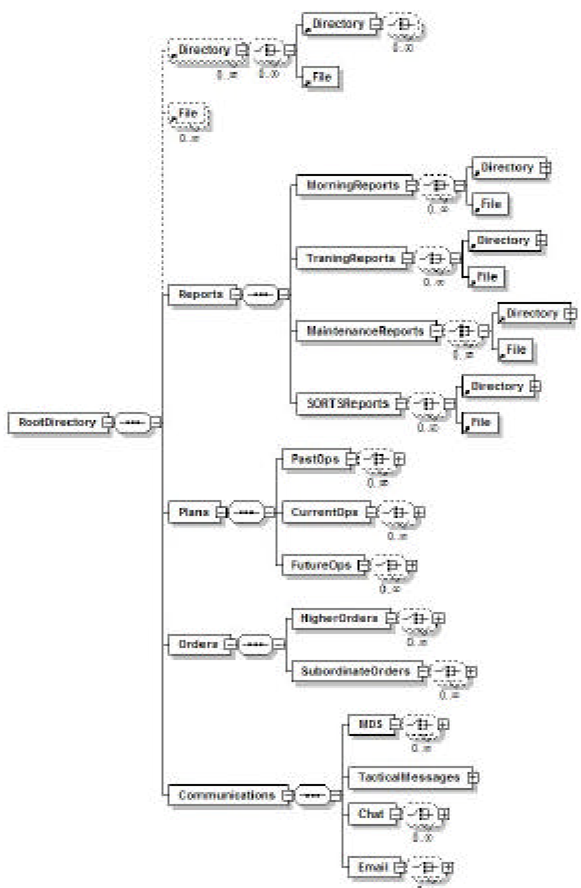


Figure 5. A Notional Unit Directory Structure

[43] *GOOGLE Search Engine*, *http://www.google.com*, Accessed: September 2003

G. CONCLUSIONS

XML applications are described as tools that leverage the XML technologies of XML Schema and XSLT to produce structured data that is validatable and can be shared across the GIG. The key distinction of XML applications is that they are data-centric, vice application-centric. The fact that applications will come and go, but that data must remain portable, and accessible, requires this approach. The role of XML Transformation for interoperability cannot be understated, since it allows integration of existing databases and systems, and promotes application independent information management at the data level.

The key to Data Control is recognizing that the most important element of information management is the information that must be managed. Structured data is the pre-eminent control measure that will allow systems independence, extensibility, and interoperability of data.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. TERRAIN

A. INTRODUCTION

This chapter introduces 3D visualization of terrain using freely available, web-browser based interfaces. The approach described models an existing terrain specification in order to selectively access large binary files without having to create redundant intermediate terrain databases. This approach diverges from traditional, application dependent approaches, and demonstrates the power of extensible methodologies.

This exemplar is available as a working web service that provides 3D views of terrain data, selectable from a 3D interface of the earth, to authorized users from any place in the world. The intent is to make this resource available to warfighters as a web based command and control tool on the local intranet.

B. OVERVIEW

Terrain visualization is an elemental concern in all aspects of combat. The ground infantryman is concerned with cover and concealment, line of sight, avenues of approach, and defensibility; all of which are terrain dependent. The communicator is concerned with signal propagation, retransmission, and antenna placement. The artilleryman is concerned with terrain masking, and trajectories. The pilot is concerned with landmarks, enemy and friendly positions, and concealed weaponry. The surface warfare officer is concerned with shorelines, bottom terrain (bathymetry) and the terrain over which guided missiles will fly.

Although tools exist that render terrain, either in 2D or 3D views, it must be recognized that the most powerful processor of military terrain related information is the human brain. Usefulness in this arena must be pragmatically gauged in relation to the degree to which human visualization capabilities are enhanced by the use of tools. A 2D map is an example of a tool that serves as a basis for planning, but does not claim to approximate reality. 3D views are more powerful because they can add on-the-ground perspective and visibility awareness that is not available in 2D. These views can, however, be misleading, inaccurate and even dangerous unless strict standards are followed to prevent gratuitous virtual reality effects that may obscure or misrepresent ground truth.

This work uses XML to adapt control measures that govern computer data processing in order to ensure data control by leadership. Military terrain data consists mainly of a National Imagery and Mapping Agency(NIMA) controlled data format called Digital Terrain Elevation Data (DTED). This data is structured, stored and delivered in accordance to a performance specification that is approved for use by all departments and agencies of the DoD.[44] This standardized format has allowed diverse systems to process compliant data files for at least two decades. The exemplar in this chapter describes the development of a Network Centric web based delivery system of 3D terrain views from a DTED terrain database. The software is designed in such a way as to be governed entirely by XML data structures that concisely model the performance specification. The intent is to demonstrate the powerful visualization capabilities that are possible, as well as the overriding importance of data control and compliance that is accomplished through the use of XML.

C. TERRAIN DATA

1. NIMA DTED, USGS DEM

File formats are a frequent source of problems among systems that require interoperability. Usually vendors maintain proprietary file formats as a mechanism for retaining their customer base by maintaining de-facto ownership of the products that their software produces. Full data control can never be achieved if dependence on proprietary file formats, and the software that is required to read those file formats, is permitted for use in the DoD. The file formats that the NIMA, and the United States Geological Survey(USGS) have mandated for terrain data are not subject to these problems, but derivative products almost always are.

[44] Department of Defense, *MIL-PRF-89020B Digital Terrain Elevation Data (DTED)*, July 2001

NIMA maintains DTED, while the USGS produces the Digital Elevation Map (DEM) file format and others such as Geographic Topological Data (GTOPO) and Digital Elevation Data (DED). The principle difference between the approaches of these organizations is the customer. USGS caters to civilian and industry demands, while NIMA serves mainly government agencies and the military. As with most file formats, there is a very basic relationship between form and function in the file formats maintained by NIMA and the USGS. The formats are very similar and it is not difficult to convert between one type of terrain data and another. The work in this thesis demonstrates a way in which XML can be used to model these file formats, so that conversions can be performed on the data level using XSLT.

2. Vector vs. Raster

Visual rendering of data on computers is done is two principal ways. Raster imagery is pixel based, and is most commonly associated with photos and 2D maps. Maintenance of color and position data at the pixel level is computationally intensive and becomes impractical for most 3D applications. Raster data is also very difficult to scale. An example of this is a road on a 2D map that looks fine from a typical viewpoint, but which at a “zoomed-in” view is hundreds of meters wide and has no relevance with regard to position or scale.

Vector data is mathematically rendered and “drawn” into a digital view. This means that it can be selectively “re-drawn” at different viewpoints to accommodate the scaling problems that accompany different perspectives. Efficiency is also gained by the reduction of data points required to render objects. A line for example, can be defined by two points, and does not have to consist of a large number of consecutive pixels. Of course, the actual rendering of the pixels must eventually occur on the screen, but this becomes a technical process that is handled very efficiently by graphics hardware. Vector data allows far more efficient and intuitive data control over graphics processes by separating the data organization from the rendering functions.

This does allow for a certain loss of control on the rendering level, and rendering procedures and mechanisms must also be carefully examined in order to ensure that information representation is accurate and reliable. Graphics rendering is beyond the scope of this work, and is not addressed beyond the recognition of this caveat for all types of computer graphics.

3. Scaled Vector Graphics (SVG)

One key aspect of vector data is that it is far more compact than raster data because far less information is needed to perform mathematical drawing than for pixel representation. This is one reason that web applications are moving toward the XML language of Scaled Vector Graphics (SVG).[45] This is an emergent file format that allows the efficient transfer of 2D vector data over the web for viewing using commonly available browser plug-ins. SVG is the logical format of choice for all 2D formats that require compactness and scalability. A transition from raster based maps to SVG based maps is a key requirement for the development of Command and Control(C2) software tools that support battlespace visualization.

4. Extensible 3D (X3D)

An important XML language for the rendering of 3D data is Extensible 3D Graphics(X3D).[46] X3D is recognized by the International Standards Organization (ISO) as an accepted format for the description of 3D information for rendering. Originally known as the Virtual Reality Modeling Language (VRML), X3D was developed by the Web3D Consortium[47] to embrace XML technology as the preferred method for defining complex data structures like those required to describe 3D scenes. Because X3D is an XML language, XSLT can be applied to transform an XML defined representation of the DTED Specification into X3D formatted data. This allows it to be rendered using commonly available web browser plug-ins.

5. GeoVRML

GeoVRML[48] is an extension to the Web 3D Specification that allows for multiple geographic projections and coordinate systems, high precision data

[45] Ibid. 29

[46] Ibid. 30

[47] *The X3D Task Group*, *http://www.web3d.org/x3d.html*, Accessed: September 2003

[48] *GeoVRML.org*, *http://www.GeoVRML.org*, Accessed: September 2003

types for coordinates, intuitive geo-positioning, and level of detail management for rendering. GeoVRML uses geodetic transformations that are based on the SEDRIS[49] geodesy code package that was developed by the US Government. For this reason, GeoVRML can be considered for operational use in the web based rendering of battlespace views. GeoVRML is a project of the Web3D Consortium, and is progressing with these efforts to become a fully X3D compliant technology. The primary mainstream open source application that uses GeoVRML is TerraVision, by SRI.[50]

D. EXEMPLAR: DTED RETRIEVAL AND 3D VISUALIZATION WITH XML AND X3D

1. Modeling the Specification

The first step in developing software that permits data control by standards and specifications is to model the governing specifications using XML. This process adds machine readability functionality to the human readable product by producing an XML Schema for verification and validation of data processes.

Much of the textual information in the DTED performance specification is not applicable for machine processing, but there are important stipulations that apply to the design and implementation of the XML constructs. These include such things as media type and file naming conventions. The XML Schema that was developed to model the DTED Performance Specification for this exemplar is focused on reading the binary data files, and does not encompass some of the metadata characteristics that are required to validate a fully compliant DTED data collection. This is recognized as a subject for future work.

2. File Organization and Access

The required locations, naming and organization of DTED files are stipulated in the specification, but rather than reflect this in the data format schema, DTEDSchema.xsd, it is reflected in a more generic directory oriented

---

[49] *SEDRIS*, *http://www.sedris.org/*, Accessed: September 2003

[50] Reddy, M., Leclerc, Y. G., Iverson, L. and Bletter, N., *TerraVision II: Visualizing Massive Terrain Databases in VRML*, IEEE Computer Graphics and Applications (Special Issue on VRML), 19(2): 30-38., 1999

schema, DataSetDirectory.xsd. Both of these XML documents are provided in the code collection. This approach was taken in order to demonstrate the way that file systems can be modeled and validated using standardized file system schemas.

As discussed in Section C of Chapter III, these XML Schemas can be designed and applied from higher headquarters, and used to automatically create compliant file systems on computers throughout the network to validate these systems and to implement updates and changes as required. Basically it is a way of making sure that all data that must be made available on the network is placed in a standardized and accessible file system on each computer. This is a centralized, enforceable way to ensure network publication and discovery of data in a Network-Centric fashion, and is an essential capability for effective organization and data control on the GIG.

3. XSLT Query Mechanisms for Database Functionality

This project uses the XML manipulation tools described in Section D of Chapter III. The tool that was used to create the XML representation of the DTED file system is named XMLDirectoryMap.java, and is designed to create an XML mapping of any file system that is compliant with the general schema, DataSetDirectory.xsd. Once the mapping is made, validation can be conducted using a more specific schema to verify correct directory and file names, and hierarchy.

The DTED files are organized in accordance with the specification because they were delivered that way on the original NIMA CD Rom media. The XML file that maps the DTED files on the network is compliant to the generic schema DataSetDirectory.xsd, and the naming conventions stipulated in the specification are used to query this file using XSLT, but there is no validation step that verifies proper naming and organization in relation to the NIMA specification. This is a required step for wide implementation of a terrain data publication and discovery system on the GIG.

Figure 6 depicts a graphical representation of the procedure for mapping a collection of DTED files on the network. These collections can reside on separate CD Rom disks, on network resources, or on a local computer. Once they are mapped, a transformation is performed using the XSLT script MakeGeoFileMap.xsl, which creates a streamlined file, GeoFileMap.xml, that is used to create 3D representations of available data.

The lower half of Figure 6 illustrates the use of another XSLT script to search the GeoFileMap.xml document in order to discern the physical location of a DTED file on the network. This XSLT query, GetDTEDDataPath.xsl, uses the parameters of Latitude and Longitude to match the location of the files as mapped in the GeoFileMap.xml document and to build the associated directory path to the corresponding data file for retrieval. The execution of these processes is performed by the XSLTransform.java class that is executed by a Servlet on an Apache Tomcat web server. This is an example of data driven software that uses generic classes to process structured data, instead of proprietary customized software that processes proprietary data formats.

XSLTransform.java is used to accomplish 90% of the data manipulation processes in this application. Although XSLT scripts that do the work are not simple, they are far easier to maintain and adjust than the traditional byte code implementations used to perform the same tasks. Customization of this application requires XSLT skills and web page development skills.

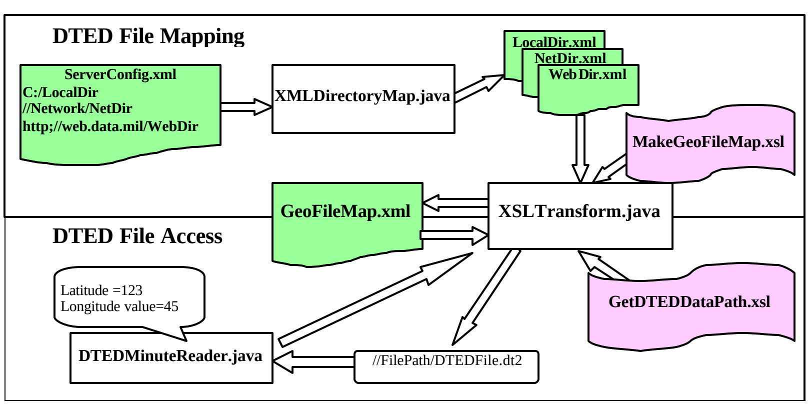


Figure 6. DTED File Mapping and Access using XML and XSLT

4. Reading Binary Data Using XML Schema

Once the DTED data is properly organized and mapped, and a file is identified, selected areas of interest can be retrieved. The amount of data that is available in an entire DTED data segment, which covers a one degree area, is not practical for rendering. It was determined that a one minute area is the most practical area segment with which to build views by tiling, because this is the next order of magnitude in the Degree-Minute-Second geodetic coordinate system. In order to do this it was necessary to write code that can use the data file format represented in the DTED Schema to selectively read data from within the main file. Figure 7 illustrates the process of reading selectively from a DTED file.

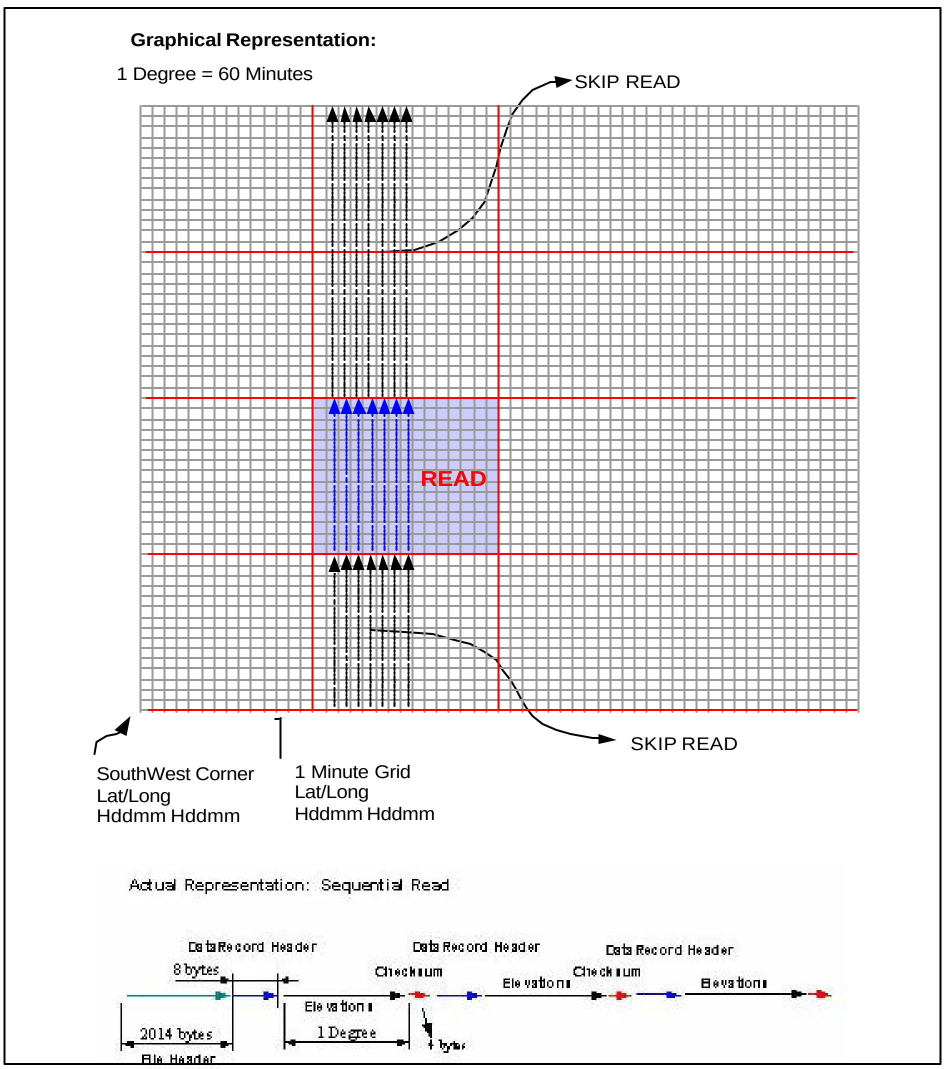


Figure 7. Selective Reading of One Minute Areas from a DTED File

The code that reads the DTED file is DTEDReader.java, which uses BitReader.java to read from the binary data. DTEDAreaReader.java manages the reading of multiple tiles, the drawing of lines, and the addition of additional X3D objects such as the location reticule. All of these utilities use the JDOM API.

5. Interest Managed Retrieval and Rendering

Interest management is a requirement that pervades the area of battlespace visualization and any software endeavor that addresses immense amounts of data. This exemplar demonstrates that a user's capacity to navigate intuitively to desired information within a huge collection can be enhanced by 3D visualization.

The design of an interface for terrain data does not require a great deal of imagination, but does offer the choice between 2D and 3D representations of the earth's surface. Attempts to represent available datasets on a 2D map of the earth resulted in extremely large windows that required scrolling and involved rendering processes that require additional software code to draw 2D areas. Because the focus was on 3D representation of terrain, it became apparent that the most appropriate vehicle for representing the entire set of available data was a 3D rotating globe. The GeoVRML model that was used for this is a 3D Globe that was obtained from the exemplars on the GeoVRML web site.[51]

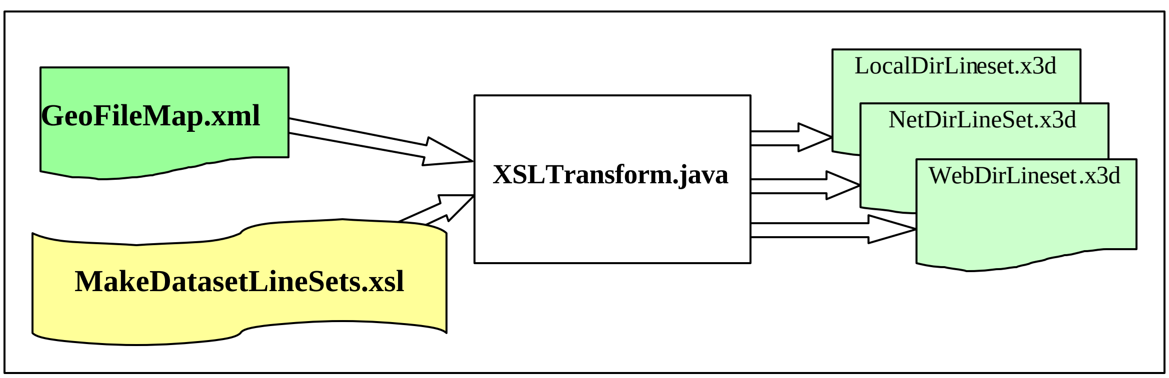


Figure 8. XSLT Transformation of XML Data to X3D files

In order to represent available data sets in 3D it is necessary to create an

[51] Ibid. 48

overlay of grid squares over the 3D globe. This takes the form of an auto-generated X3D IndexedLineSet.[52] Figure 9 Illustrates the process of generating a 3D line set representing available data. Each data set is mapped separately so that they can be viewed separately in the user interface.

Figures 9 and 10 show the constructs generated using the XSLT transformation, MakeDatasetLineSets.xsl, that operates on the GeoFileMap.xml file. This is a transformation that converts an XML file containing geographic data into an X3D file that draws lines to represent the data in 3D. Each square represents available DTED Level 1 data.

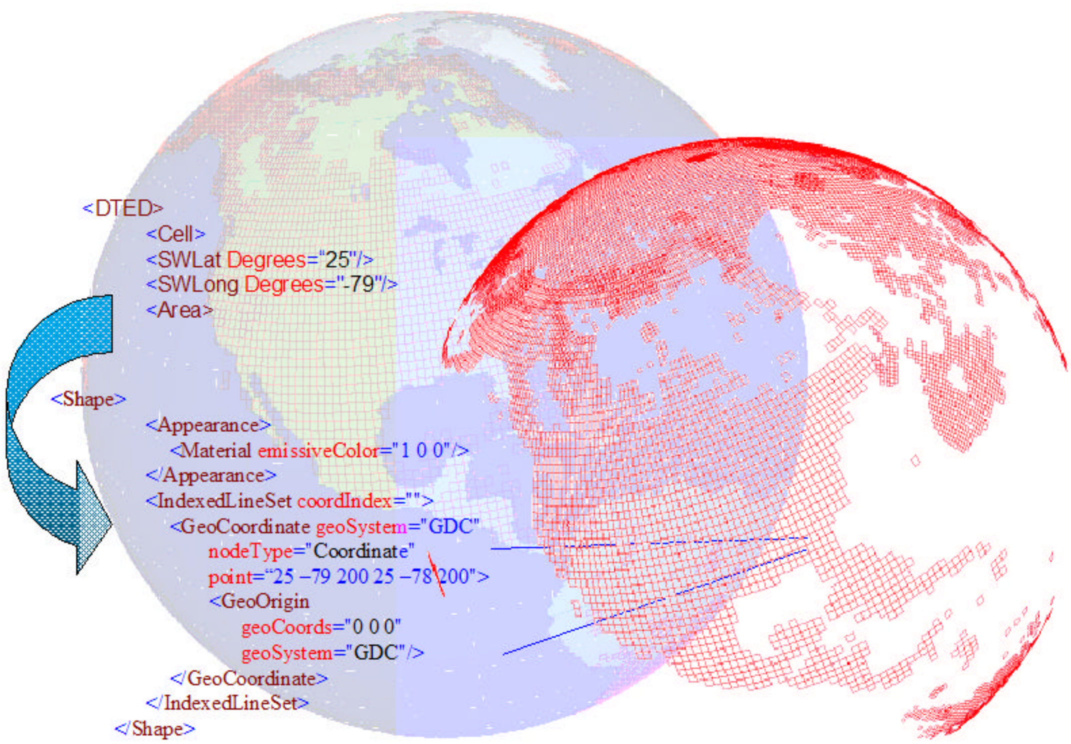


Figure 9. Generation of a 3D Line Set Representing DTED Data

52 Ibid. 47

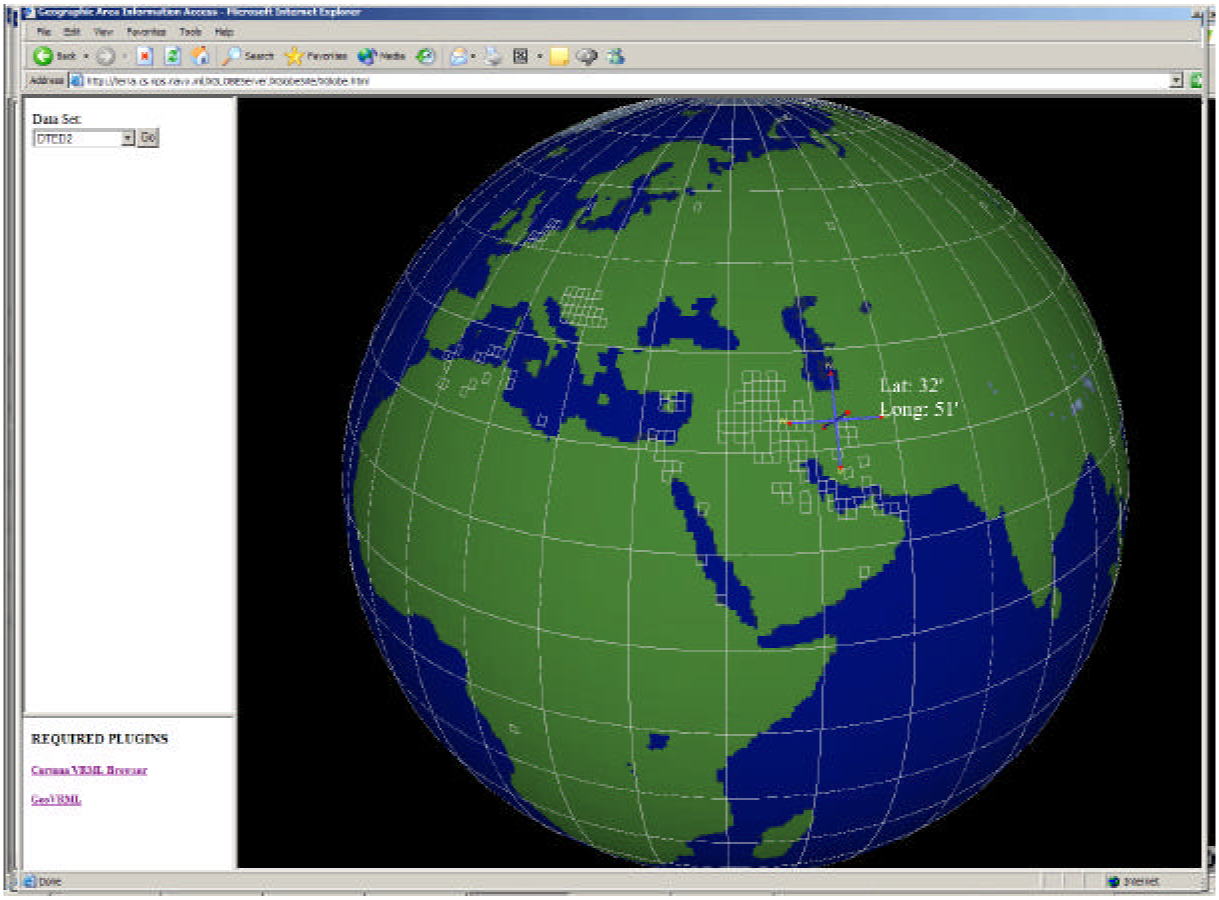


Figure 10. 3D Line Set Representing a DTED Data Collection on the Globe

6. Scene Generation

The XGLOBEServer application that is provided in the software collection allows the selection of any dataset that it is initialized with. The selection of the dataset will change the overlay to reflect available data by applying an XSLT transformation to the underlying data and refreshing the view. The same transformation tool that creates the line sets and retrieves file data is used to perform the transformation that alters the overlay. Most of the operational details of the interface for scene generation and retrieval is handled by XSLT instruction sets that are processed by a generic Java Servlet on the Apache Server. This methodology can also been applied on the local machine, but because the browser requires a server mechanism in order to access the local file system, it is necessary to instantiate a server process locally.

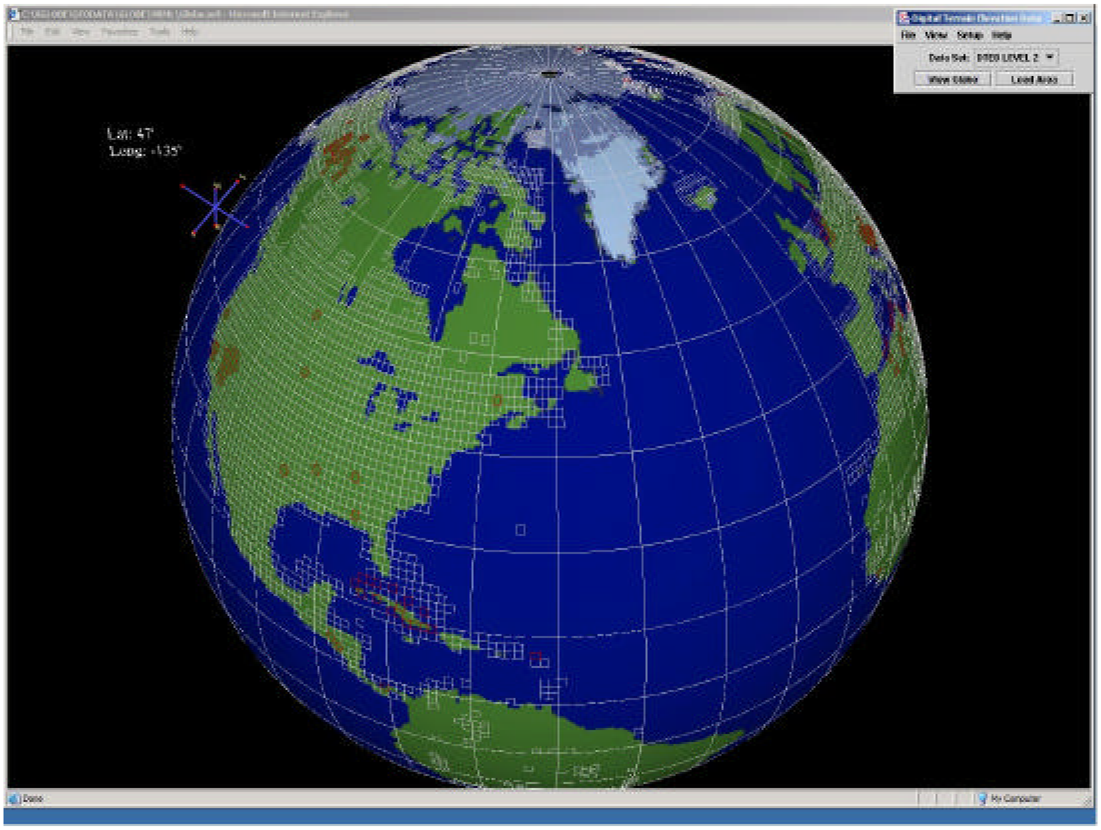


Figure 11. Globe View Showing DTED 1 and DTED 2 in Red

Figure 12 illustrates the process of reading the DTED data, and creating a scene which tiles together several one minute DTED X3D files, as well as adding gridlines and a reticule for displaying location, is the most code intensive operation in this application. DTEDReadServlet.java uses XSLT processes to extract information from the index data, and calls DTEDMinuteReader.java, DTEDAreaReader.java, and TerrainGrid.java Java classes to write the data to X3D template files. DTEDMinuteReader.java calls the BitReader.java class to load the DTED Schema based template file and read the binary data sequentially into the elements by referencing the “bitLength” attributes in the XML.

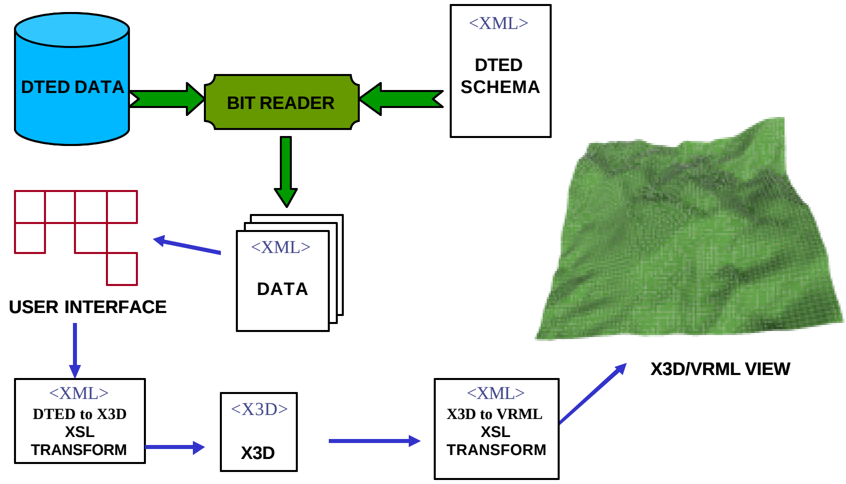


Figure 12. DTED View Generation Process

Early versions of this project used data from single DTED CD Rom media, and a standalone Java application. A screenshot of this application is shown in figure 13. This application is limited in that it can not allow selection from an entire DTED collection, and it does not tile multiple one minute areas together. The blue areas in the figure show the available data on a particular DTED disk.

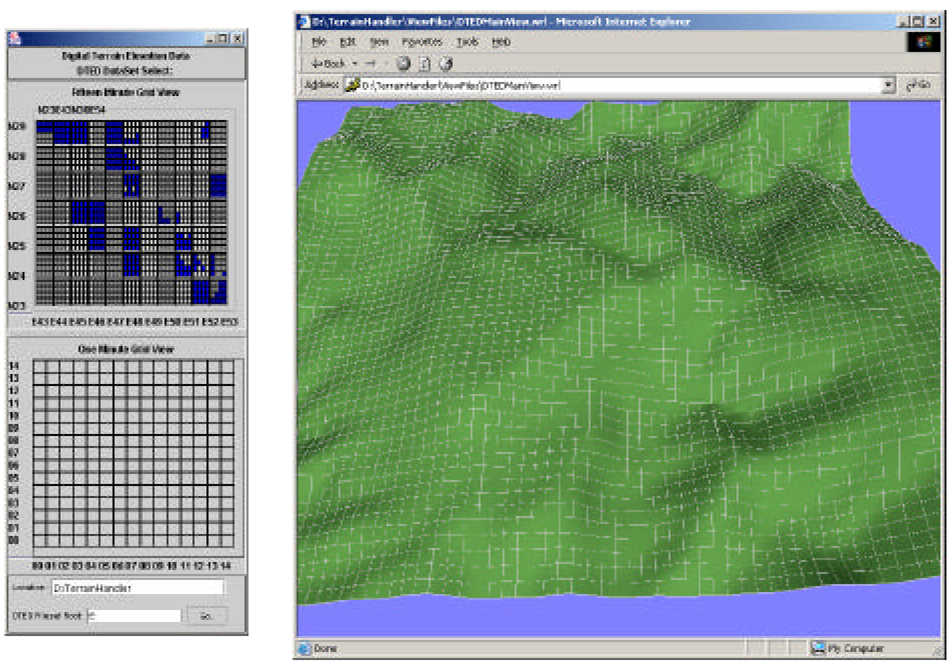


Figure 13. Early Version of Terrain Extraction Application

Figure 14 is a screenshot of the final browser based selection mechanism for selecting an area of interest on the one minute scale, once a degree grid is selected. The grid depicts a one degree area. A selected minute is the lower left hand corner of the terrain to be loaded. The application defaults to a 5X5 minute grid area. The application is a web service, so this browser interface is not necessary to retrieve terrain. A direct Java Servlet Query of the form:

```
http://terra.cs.nps.navy.mil/XGLOBEServer/
servlet/XSLTServlet.DTEDReadServlet?
LatDeg=32&LatMin=46&LongDeg=48&LongMin=40
```

can be used by any web browser or application in order to retrieve terrain data for viewing. A potential application of this approach might be to speak a unit position into voice recognition software, have it generate the above URL using voice recognition software, and load the appropriate terrain view in the browser. This will provide immediate visual representation of a position report, or contact report. This is reserved for future work.

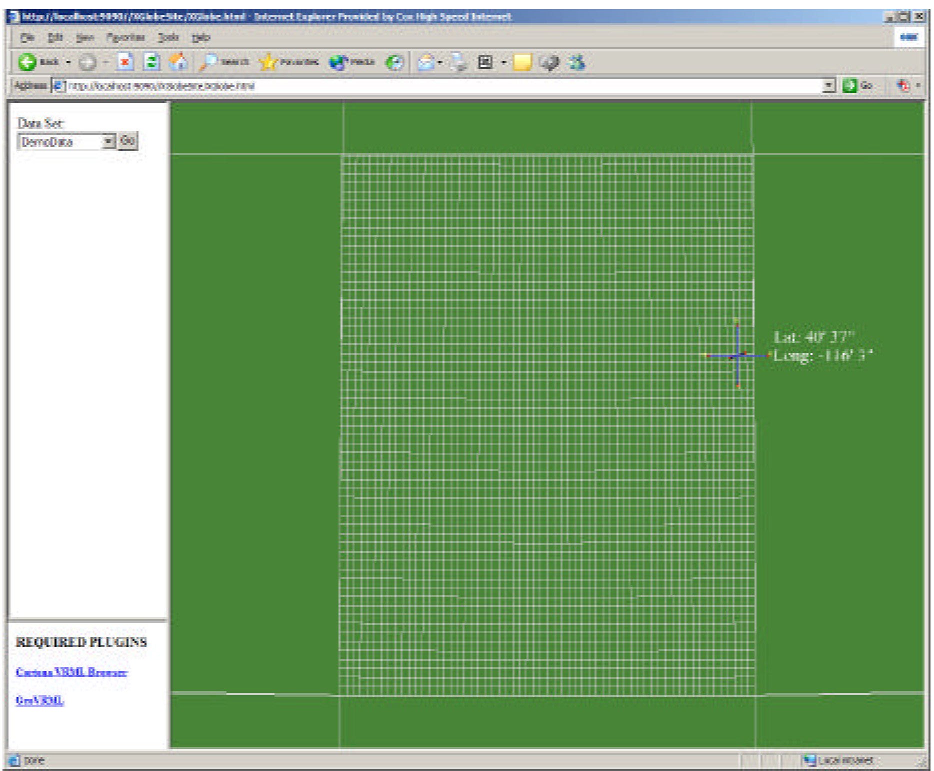


Figure 14. Degree Grid Selection Screen for Area of Interest

Figure 15 shows a portion of terrain extracted using the final version of the software. The geo-located markers serve to demonstrate that actual location information Is rendered using this software, and the ability to create markers demonstrates the function of placing and moving units in the scene.

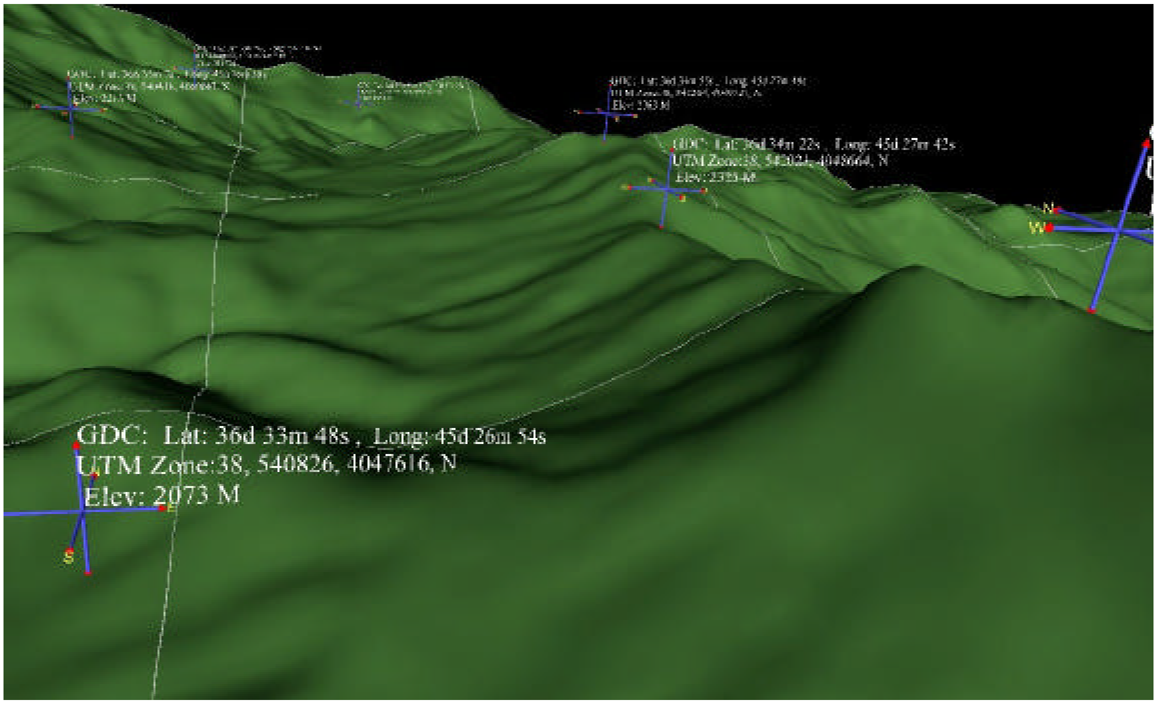


Figure 15. Terrain Extracted From DTED File.

The terrain server application developed in this work is currently available on the internet, so that any DoD user can access and download 3D terrain, as well as the source code for this implementation of XML driven technology. Because DTED data is Limited Distribution (LIMDIS) a password is required to access the site. Instructions on how to obtain a password and access the site are given in Appendix A.

7. Application Data Access

Rendering applications will always require specific adaptations. The fact that high level 3D rendering can be accomplished within web browsers using widely available plug-ins is an indicator that robust Network-Centric technologies are no longer dependent on complex “thick-client’ installed applications. It is of course preferable to develop military specific browser plug-ins that are not dependent on proprietary software and that accommodate the specific needs of warfighters. A potential project that will support this is XJ3D[53]., which is the

[53] *XJ3D*, *http://www.xj3d.org/*, Accessed: September 2003

reference implementation for the X3D language. All the examples rendered in this exemplar are accomplished using web browsers and a freely available browser plug-in.

There is no reason that standard, locally installed applications cannot access web resources and render 3D terrain as well. As storage capacities rise it is feasible that every military computer will have rapid local access to all of the necessary terrain and imagery data necessary for current operations. The path to this level of data availability and accessibility is standardization of data formats and software that is designed to accommodate that data.

8. Battlespace Aware Scenes

Because web browser technology is already uniquely suited to Network-Centric endeavors that involve the manipulation of data from external sources, they represent a logical starting point for the development of scenes that can update themselves in response to published network events.

Figure 16 is a screenshot of a portion of terrain that was populated with a tank during a Joint Conflict and Tactical Simulation (JCATS)[54] simulation during a Distributed Continuous Experimentation Exercise (DCEE).[55] To make an X3D scene "Battlespace Aware" requires code that can react and respond to network messages. The project uses a software mechanism to translate High Level Architecture (HLA) simulation messages to X3D for rendering and movement in the scene. The method uses existing 3D tank models, from the Naval Postgraduate School (NPS) collection of 3D models in the NPS MOVES Institute[56] Scenario Authoring and Visualization for Advanced Graphical Environments (SAVAGE) library[57] and positions them according to their published locations on the terrain.

---

[54] *USJFCOM Joint Conflict and Tactical Simulation(JCATS)*, *http://www.jwfc.jfcom.mil/about/fact_jcats.htm*, Accessed: September 2003

[55] *USJFCOM Distributed Continuous Experimentation Environment(DCEE)*, *http://www.jfcom.mil/about/fact_dcee.htm*, Accessed: September 2003

[56] *Naval Postgraduate School MOVES Institute*, *http://www.movesinstitute.org/*, Accessed: September 2003

[57] *Scenario Authoring and Visualization for Advanced Graphical Environments - SAVAGE Library*, *http://web.nps.navy.mil/~brutzman/Savage/contents.html*, Accessed: September 2003

Using the techniques described in this work, exemplars that are extant on the web, and functions that can be reproduced using the provided code, warfighters can view distributed simulation exercises, or real time operational situations as long as data control is maintained, and basic Network-Centric data dissemination principles are followed. The JCATServer package that is included in the code set is an adaptation of the XGLOBEServer code that creates battlespace aware scenes. This is a starting point that is being leveraged by the Extensible Modeling and Simulation Framework (XMSF)[58] project which is spearheaded by the MOVES Institute at the Naval Postgraduate School.

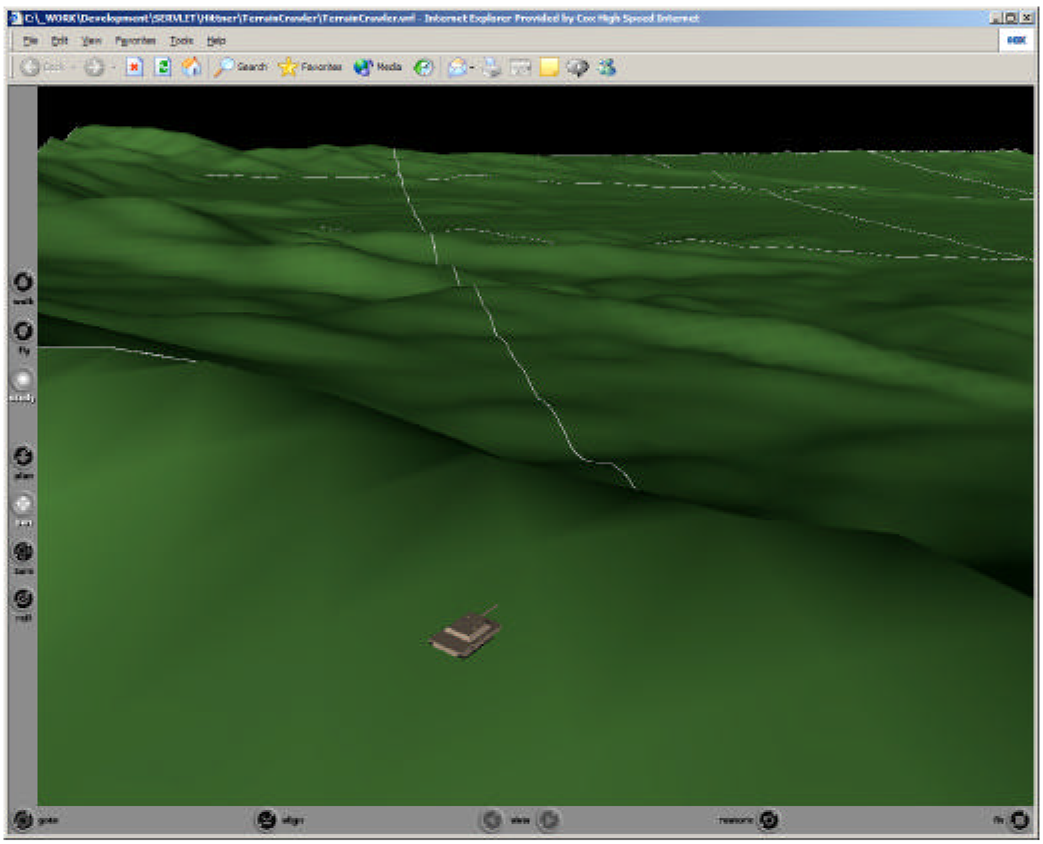


Figure 16. Tank on Terrain During a JCATS Simulation Exercise

9. Network Delivery

Conventional wisdom relegates 3D terrain rendering to standalone applications that require great processing power, skilled operators, and a considerable amount of time. Most terrain imaging products create scenes by

[58] *Extensible Modeling and Simulation Framework (XMSF)*, *http://www.movesinstitute.org/xmsf/xmsf.html*, Accessed: September 2003

manually loading terrain data in. This usually requires a trained operator. This is impractical for use in the Command and Control environment, and does not effectively leverage battlespace information. The main reason for this approach is that the file sizes are too large for network delivery. This is why the selective read approach was taken for this project.

A one-minute square area of DTED 2 terrain, with one posting every 30 meters can be placed in a 23 KB XML file. When this is compressed it takes up 4KB. Because computers can be pre-populated with model libraries and other graphics and imagery that may consume bandwidth, there are few other bandwidth intensive requirements. If a 5X5 minute area can be delivered in a file that has a compressed size of 100 KB, it becomes very clear that network delivery of web based 3D terrain is not impractical, and that there is no reason why this capability cannot be placed in the hands of warfighters at all levels.

This exemplar demonstrates auto-generated, and auto-populated 3D views of the battlespace. The advantage of being able to perform fly-throughs, and get a bare earth perspective on a location before having to go there is inestimable. If current C2 tools can be aligned with this paradigm, then 3D web views of the battlespace can be network deliverable, customizable, and easy to access and use. Commanders and operators can elect to view an area of interest in 3D by opening a web browser.

10. Data Control Issues

Because terrain visualization is currently "product-based" and is not broadly distributed as a general information resource, there is a great deal of variance on how file size, resolution, rendering, viewing and navigating are handled. In order to ensure consistent treatment of terrain in modeling and simulation and for use in command and control tools, it is important to establish a baseline, validatable 3D format that can be the basis for all rendering processes. This work proposes that the open source, open standards solutions that exist in X3D and GeoVRML are the most Network-Centric and extensible solutions.

Application centric, product based solutions are currently being developed for use DoD wide in the Commercial Joint Mapping Toolkit(C/JMTK).[59] These products achieve interoperability and Network-Centricity only through artificial internal extensions to the closed source Microsoft™ COM architecture. There is great potential for leveraging the capabilities of the NIMA developed JMTK in conjunction with the technologies that are proven in this exemplar, but there is very little potential for the C/JMTK because of its stove-piped architecture, proprietary file formats, and lack of 3D support.

Figure 19 compares he C/JMTK architecture to the methodology applied in this work. The critical reader will point out, as the C/JMTK developers do, that the solution on the left is 90% operational, while the extensible solution is not. The work demonstrated in this exemplar shows that direct source data access and rendering can be achieved on a high level that in fact surpasses current tools in terms of accessibility, network delivery, and 3D functionality. The solution on the left requires a collection of applications, components and database mechanisms that are expensive and difficult to implement. The solution on the right requires a web browser, a web server, and NIMA DTED data, all of which are freely available.

E. EXTENSIBILITY

This project address the problem of data accessibility, and visualization. In this context, the term accessible applies to information that can be provided by a networked server. It is not enough to simply provide a view of this data, as many of the proprietary solutions can already do, but this data must be in a format that can be further processed by applications that include more battle space information. The requirement is to model a large set of large files in such a way that manageable portions can be extracted and delivered efficiently and in a useful format. This solution models DTED information using the XML, extracts

[59] *Commercial Joint Mapping Toolkit (CJTK)*, *http://www.cjmtk.org*, Accessed: September 2003

requested data in manageable portions, and delivers the data in an XML format that can be transformed immediately and viewed in a standard internet browser viewer.

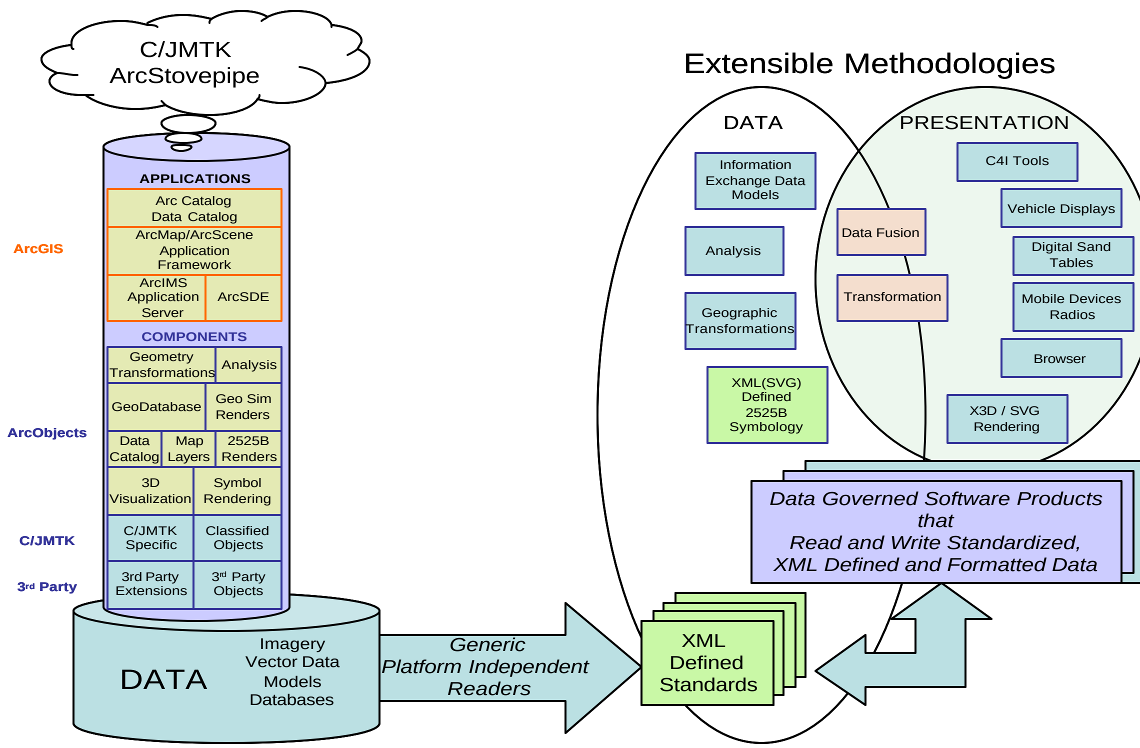


Figure 17. The C/JMTK Compared to Extensible Solutions

This project also demonstrates the treatment of computer file systems and the data that they contain as database resources in themselves. This is, perhaps, the direction in which Network-Centric database technology is moving. The creation of applications that simply contain data within the larger framework of the network and internet will eventually be recognized as a focus of needless duplication of human effort and processing power. This program demonstrates one way that data can be modeled in a machine readable format and can be accessed and delivered independently of traditional database engines. In the future, data will not be made available by being placed in a database, but rather it will make itself available to applications by complying to certain formats or by providing its own schema that will allow transformation to a common thread.

A motivator for development of this software was that available commercial products are expensive, “product-based” and use proprietary file formats that deny the possibility of Data Control. It is hoped that the methods and principles demonstrated in this exemplar will serve to instigate rebellion against solutions that promise extensibility, but deliver application-centric systems that are designed as much for self perpetuation as for functionality.

F. CONCLUSIONS

This project is a success on many levels. 3D views of DTED data are now available to authorized users with a web browser. Warfighters in all theatres can use this to examine terrain that they will have to fly over, patrol, or otherwise exert control. Command and control tools can extend this process to place friendly and enemy units on the terrain, and to update the view in response to live report entries.

Infrastructure requirements for implementing this tool at a unit, or on a single machine are minimal, especially when compared to the multitude of servers and applications that are required for current systems. A Java based web server is all that is required. The organization and delivery of information from large binary files does not require expensive and complicated intermediate database systems, and all data retains the integrity and structure that is defined in the DTED specification. This project achieves Net-Centricity, interoperability, and Data Control.

THIS PAGE INTENTIONALLY LEFT BLANK

V. GEODESY

A. INTRODUCTION

This chapter addresses the area of geodesy and the need to institute a consistent and interoperable method of position reporting. The problems associated with reliable position reporting, and interoperability are discussed. The application of Data Control to ensure that position reporting systems are interoperable and compliant to the needs of the warfighter is proposed. The exemplar presents an XML Schema that defines the information that a position reporting system must process and provide in order to participate on the Network-Centric battlefield.

B. OVERVIEW

Geodesy is the practice of reconciling the irregular shape of the earth with mathematical models that allow the precise location of objects and areas. This is of course, essential to all modern military operations and has been a mainstay of military science for centuries. The process of terrain visualization demonstrated in Chapter IV is highly reliant on geodesy tools to ensure accurate placement of terrain on the earth, and of objects on the terrain.

The calculation of global location is not a simple endeavor and requires the reconciliation of asymmetric global coordinate systems, such as Geodetic Coordinate System (Latitude and Longitude), and symmetric local coordinate systems, such as the Military Grid Reference System (MGRS) which is based on the Universal Transverse Mercator (UTM) system[60]. Battlespace information systems must utilize a common, tightly controlled toolset in order to ensure that consistent and accurate geodesy is maintained. Algorithmic differences between software products that provide geodesy support cannot be permitted.

In order for military leadership to take full responsibility for this vital fact and to be assured of software compliance across the board, it is imperative that standard conversion mechanisms be established and enforced. These

[60] Department of Defense, *Joint Publication (Joint Pub) 1-02, "Department of Defense Dictionary of Military and Associated Terms"*, April 2001(As Amended Through June 2003)

mechanisms can be expressed in human readable XML Schema formats. Software can be required to validate calculations against this common format. For example, a verifiable mechanism must be established by which software tools can choose the same projection in a given location, so that software tools will not be able to introduce errors of inconsistency that can prove disastrous in a collaborative C2 system. All expressions of location must include appropriate geodetic information. It is the responsibility of leadership to define and publish what this information must be.

The most pertinent use of Geodesy in military operations is in position reporting. There are few battlefield entities that do not either need to report their own position or receive accurate reports on friendly and enemy positions from a wide range of applications. Position reporting procedures by software requires standardization and Data Control if systems are to be interoperable.

C. GEOGRAPHIC LOCATION

1. Projections and Conversions

In order to use trigonometry and calculus for position determination, triangulation, fire support, air support, and many other important processes in war, it is necessary to use a notional uniform grid that approximates the local surface of the earth as flat. This is accomplished using a geographic methodology called projection. Because the earth is not a uniform globe, projections vary for different locations on the earth, and have different degrees of distortion depending on the underlying mathematics and approximations. It is important that battlespace visualization software have the capability to reconcile appropriate projections.

In order to convert from one coordinate system to another, various factors must be known. One of the functions of structured data is that it includes information that will allow its proper use in a specific context. A schema that pertains to geographic location must include all of the projection and coordinate system information necessary to perform consistent and accurate conversions that are independent of the software that uses the data.

2. Consistency and Validity

The use of XML Schema to govern data that contains geographic information is an important step in obtaining and maintaining control over these important software processes. As many and varied systems combine to create common battlespace representations it is important that all geographic location information be properly structured and validatable in order to prevent inconsistencies and errors.

3. Joint/Combined Operations

Often the important mathematical functions that are required in the interpretation of geographic location data are left in the realm of proprietary software and file formats. This means that in joint and combined operations, the only way to have consistent conversions is to make sure that each organization uses the same software suites. This is impractical because of the expense, the lack of resources in some countries and organizations, and not least because even within the realm of proprietary software there are versioning and incompatibility problems that interfere with interoperability.

International standards have long been a mainstay for industry in the rectification of these important compatibility problems. Requiring that all software use XML Schema validatable International Standards Organization(ISO)[61] standards in the way that they publish and utilize geographic location data is far more feasible than requiring conformance within a particular proprietary solution. The fact that these ISO standards can be expressed in human readable form also gives leaders and program managers a way to guarantee that conversion methods are compliant.

4. Position Reporting

Current solutions which enable automated position reporting are systems oriented and have no exposed, standardized data format that independent software tools can use to display the data. There is little or no guidance or

---

[61] *International Standards Organization(ISO)*, *http://www.iso.ch/iso/en/ISOOnline.frontpage*, Accessed: September 2003

leadership control to dictate standard formats for position reporting, even though the priority level is high, the potential for error is high, and the cost of failure is human lives.

5. XML Schema Solution

Section C of this Chapter presents a notional XML Schema that represents a starting point for a position reporting data format. It is purposely simplified, and contains only the minimum amount of information by which any device might report its location using the required meta data of geographic projection and coordinate system. Some optional fields are included in this data model in order to allow devices to report a contextual position, such as a building, and to report a position relative to an already reported contextual position, such as a room in a building. Another field allows the entry of a radio frequency that can be used for triangulation purposes. A responsible command decision to exert control over all position reporting software is to simply require that all software be capable of reading and producing data that can be validated by an XML Schema like this one. This way vendors will simply not be permitted to write software that does not comply with interoperability and data content standards that are defined by leadership.

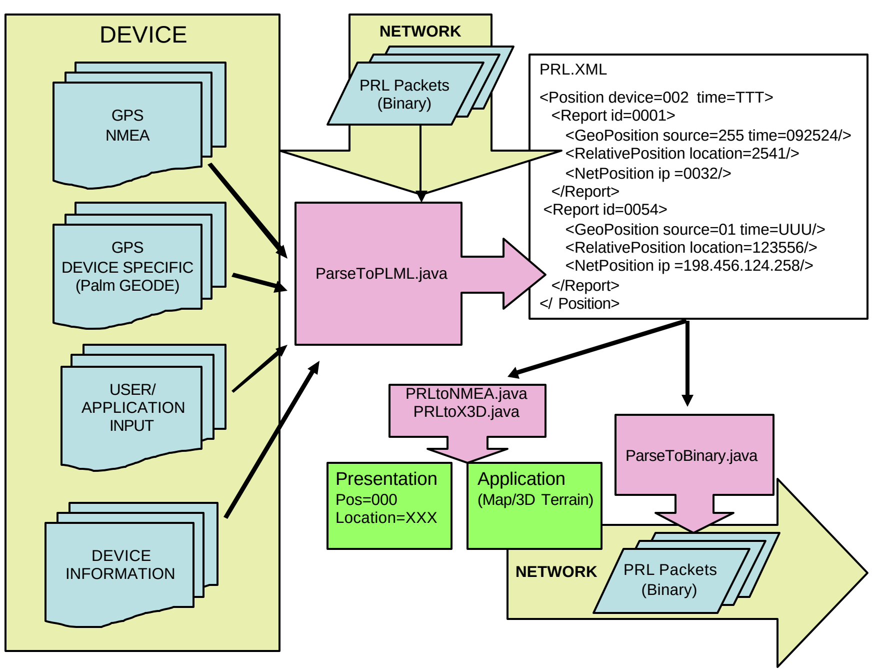


Figure 18. Position Reporting Language Processing

6. XSLT Conversion

Section C introduces an XSLT and Java application that demonstrates how a basic position reporting transaction might occur between two pieces of software. Figure 19 illustrates the process. Because most position reporting is based on GPS readings, a Java conversion is provided that converts GPS binary data to the Position Reporting Language (PRL). This is then transmitted to a hypothetical tactical device that requires data in UTM format, vice the Latitude and Longitude that the GPS provided. Because the initial message contains the requisite coordinate system and projection metadata, and because the code is XML Schema aware and is able to follow these metadata instructions, the receiving program is able to convert the GPS report to a Grid Coordinate. The result of this is that the receiving program is able to apply basic trigonometry in order to calculate the distance between itself and the sending unit.

The importance of this exemplar is not that some code was able to do conversions or calculate a distance, but rather that it was able to do so in response to information received that adhered to a common XML Schema defined standard format. It is also important to note that this methodology does not require that XML text be sent in the communications portion of the transaction. A compact binary delivery is preferable, but it must be converted to XML on arrival in order to be properly validated and interpreted.

D. EXEMPLAR: POSITION REPORTING MARKUP LANGUAGE

1. Introduction

In the recent conflict it became immediately apparent that the various tools in use to track friendly forces were not compatible. The remedy was to adopt a single solution. This is a common event in military adoption of new technologies, but it remains to be seen if the new tool will be yet another stovepipe system that will represent a future compatibility hurdle.

It is clear that for the critical and highly volatile mission of force protection that mobile technologies must be applied in a cohesive manner. Position location functions must be ubiquitous and reliable so that all possible assets can be applied to keeping our personnel safe from the considerable firepower that we produce. If we are to learn from the mistakes of the past, so evident in the recent conflict, we will address the problem by asserting some basic requirements on developers and integrators.

Markup languages fill a critical gap between human readability and software functionality. Leadership and subject matter experts can agree upon requirements and implementation without requiring a great deal of computer programming knowledge. XML is specifically designed to accommodate change over time and to separate presentation from data. This means that consistency can be achieved on a physical and data exchange level, while flexibility can be applied at the interface level to accommodate the wide variety of devices that must be brought to bear.

2. Current Technology

The current state of affairs with regard to position location functionality for mobile devices is marked by a tight coupling between hardware and software functions. Predominant standards include the National Marine Electronics Association specification (NMEA-108) and the Rockwell proprietary standard. These are directly tied to transmission protocols and device specific functions, and fail to define a single, cohesive set of requirements that position location software can be designed to fulfill or address.

There are various solutions that are being advanced in various areas. The Location Reference Message Protocol[62] is a packet based system proposed by Goodwin et al. to support transit related problems. This approach also combines transmission protocol requirements with information requirements. This approach serves certain technologies, but excludes others. For this reason it is not appropriate for an overarching solution.

An approach that recognizes the need for a cohesive encapsulation of

[62] Goodwin, Cecil W. H., Gordon, Stephen R. and Siegel, David, Re-interpreting The Location Referencing Problem: A Protocol Approach, Proceedings of GIS-T 95 Symposium, Washington DC: American Association of State Highway and Transportation Officials (AASHTO), 1995

data functionality in order to promote interoperability is the Generic Positioning Protocol (GPP).[63] This approach seeks to combine principles advanced by NMEA, the Geography Markup language (GML), the Ericsson Mobile Positioning Protocol, and others. In the referenced paper the requirements for an XML based GPP are outlined, but there is little indication that it has been implemented on a large scale.

3. Interoperability Solution

The Position Reporting Language (PRL) that is proposed in this exemplar is an extremely generic starting point that seeks to define the baseline requirements for position reporting from which many factors can be calculated. As an extensible language, the purpose of PRL is to provide a lowest common denominator of information exchange functionality. Notable functions like bearing reports, systems characteristics, reliability estimates, and many other functions that can be found in the NMEA-108 protocol messages are not included in PRL, since many can be included as extensions of the base data model, and can be calculated by software.

The purpose of a markup language is not to apply limitations, but rather to define minimal functionality to ensure interoperability and extensibility. The military is an appropriate venue in which to take the lead on data control in this area, since the stakes are so high. A baseline set of data standards must be defined and reflected in development requirements if we are to escape the problems that stovepipe technologies and proprietary solutions cause. PRL is a starting point for this.

The development process of PRL in this project transcended the scope of pure position location reporting and raised many important issues that resulted in redesign and alterations. It became apparent that such things as reporting areas of interest, the tracking of reports, and network loading issues must be accommodated. Change is inevitable. The process of defining, maintaining, and

[63] Nord, James, Synnes, K°are and Parnes, Peter, An Architecture for Location Aware Applications, 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9, January 2002

extending PRL must recognize the inevitability of change. XML based applications must accommodate this change by linking functionality to the PRL schema, and by applying XML transformation to achieve separation of presentation and data.

Position locating devices will take many forms and perform in many areas that are as yet un-anticipated. It is important to recognize and leverage existing software and data design technologies to ensure that these capabilities are maximized in support of military operations.

4. Design Issues

The initial version of PRL is deliberately simplistic, and addresses two conceptual types of position reporting. In order to be useful, and to prevent unnecessary traffic, it is important to maintain a context capability in data structure design. This means that it is not required to transmit absolute locations once a location context has been reported and can be referenced. For example, once a building's location is reported, subsequent reports can just specify a room or area in the building.

The way that relative areas are to be defined and reported can be left up to applications. The PRL architecture simply provides a consistent place in a message where an application can expect to find such data. Contextual definitions in PRL include designation of geographic system used, physical street address information, network address information, and network characteristics such as timing and frequency to be used for triangulation in a wireless environment. The Data branch of PRL is meant to contain absolute data that defines location and a reference to an established context. Again, the use of these fields will be application specific, but the semantic expression of the data will be consistent.

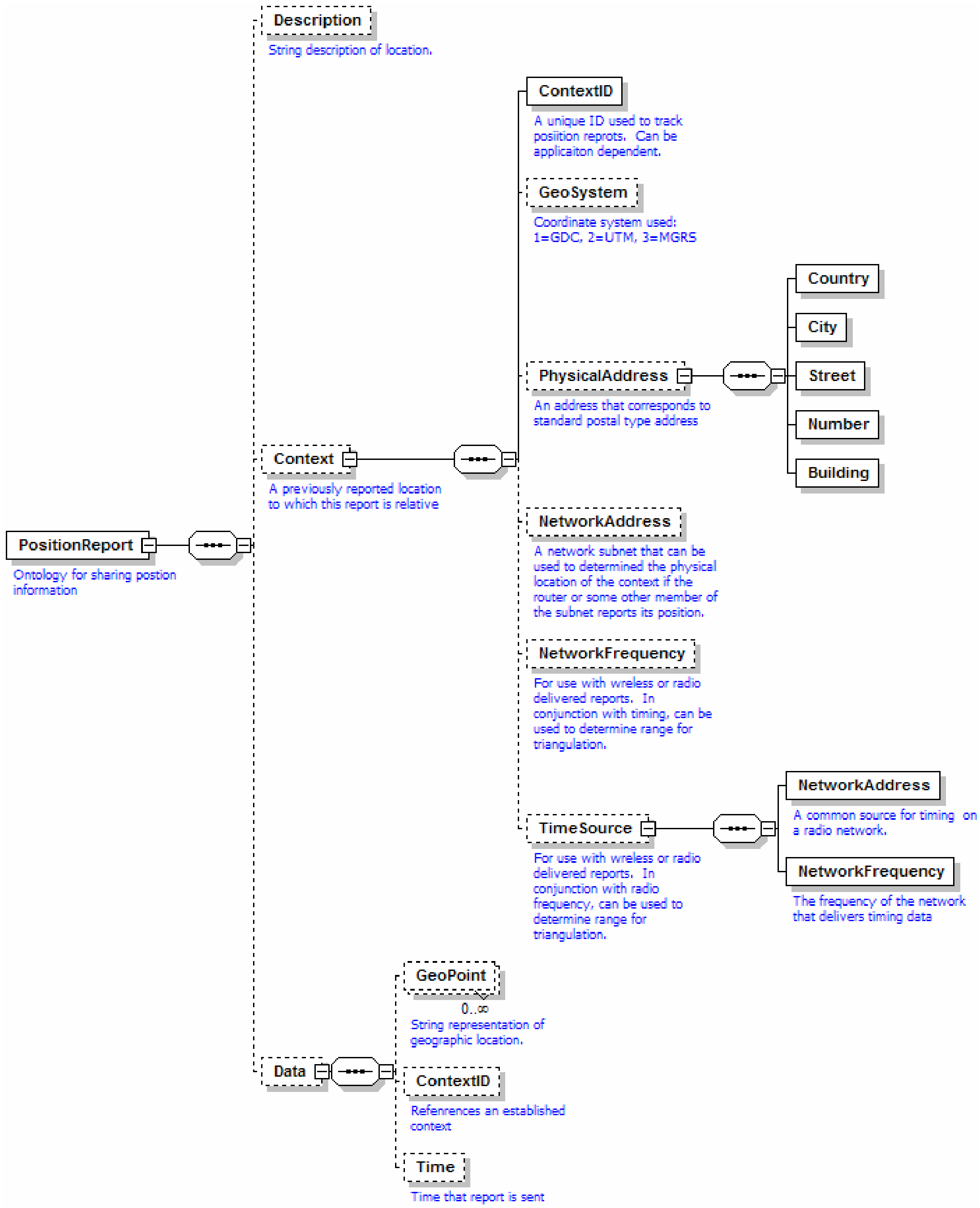


Figure 19. Position Reporting Language XML Schema Architecture

5. Implementation

Data transfer, compression, network considerations, and application specific factors such as data storage and queries must be abstracted from the XML representation of position data that PRL represents. These factors fall under the category of data presentation.

In the portable computing environment it is seldom practical to send XML data as plain text. A template-to-receptacle approach is probably appropriate in most cases, with some form of optimized binary packaging applied in between for transmission. These methodologies must be applied with the overriding requirement that data will always be expressed as XML on the application level. Figure 20 illustrates the process in which the only place that PRL exists as actual XML is in the center box. This is the Data Control area in which interoperability is specified and mandated. It is the responsibility of applications to transform to the standard.

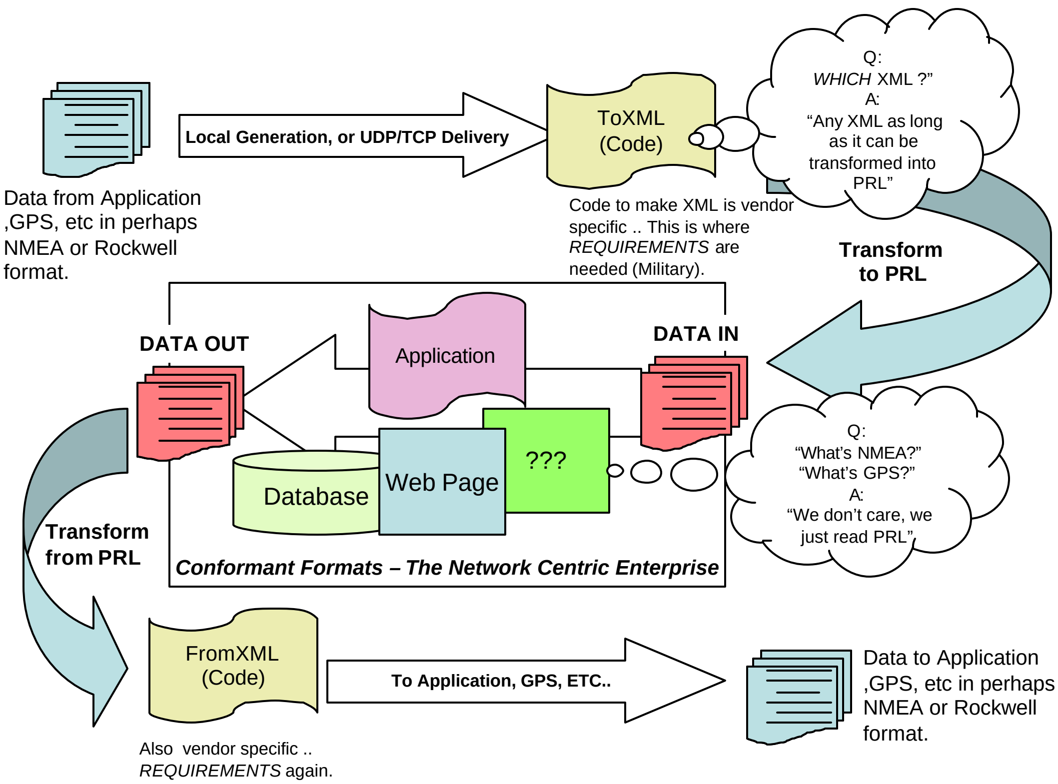


Figure 20. PRL Processing

6. Results

The utility of a consistent data format is evident in the way that this exemplar was accomplished with a minimum of data structure related adjustment. Different interpretations of the PRL data model presented some significant problems during development, and there were some difficulties with separating the roles of the data model from the roles of the applications.

It becomes very apparent that explicit documentation and implementation recommendations are imperative for the use of XML based technology to obtain data control and interoperability. Initial efforts result in fundamental changes to the initial PRL design, and make it clear that the maintenance of such a model will be an ongoing process that will require oversight and objective management.

E. CONCLUSIONS

Computer implementations have made the mathematics associated with geodesy almost transparent. The lack of interoperability between hardware and software tools that perform geographic position determination is inexcusable and is an indicator that there is no Data Control by leadership in this vital area. Demanding that all software use common data formats, and resolving to specify those formats, is a bold and disruptive approach to a very real problem. The reason that this can be considered disruptive is that it challenges some very successful profit models that industry maintains in its relationships with the military customer. Embedding software within systems, and selling the package is a profitable business model.

To require Data Control, and to impose these requirements on all military software does not disrupt any success models with regard to functionality. In this regard it is not disruptive at all, but is merely a common sense approach to dismantling what is essentially a compounding and recursive failure model that is caused by a system of incompatible systems. Compatibility is not a matter of conformance to common applications and tools by people and organizations, it is a matter of application conformance to data structures and interoperability requirements that are defined by people and organizations.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. BATTLESPACE INFORMATION EXCHANGE

A. INTRODUCTION

Battlespace visualization encompasses far more than accurate representation of the physical battlespace. A huge amount of data is contained in plans, operations orders, situation reports, tables of organization and all of the various ways in which military entities communicate. Because joint and multinational operations are the norm rather than the exception, information exchange poses a significant interoperability challenge.

This chapter addresses ongoing efforts to insert data control measures into the operational environment in order to achieve interoperability. These standards are intended to define information models to which all systems will be required to publish and subscribe. The information models, or ontologies, are designed to allow database interoperability and information exchange by establishing agreed upon definitions for most forms of information exchange that occur in the battlespace.

B. THE COMMAND AND CONTROL INFORMATION EXCHANGE DATA MODEL (C2IEDM)

The Command and Control Information Exchange Data Model (C2IEDM)[64] is based on the concept of a Battlespace Generic Hub (BGH), as illustrated in Figure 21. This approach applies broad categories to military activities that can be applied across the spectrum of joint and multinational forces.

The C2IEDM is under the cognizance of the Multilateral Interoperability Programme (MIP)[65], which leverages the Army Tactical Command and Control Information System (ATCCIS). The C2IEDM is a NATO led initiative to standardize the way that information is exchanged between international and

---

[64] NATO, Army Tactical Command and Control Information System (ATCCIS) Working Group, "*The Land C2 Information Exchange Data Model,"* Edition 5.0, ATCCIS Baseline 2.0, March 2002 Recent consensus has agreed to drop the 'Land' in the Title.

[65] Multilateral Interoperability Programme(MIP), *http://www.mip-site.org* Accessed: September 2003

joint Warfighting communities. This effort follows the NATO Code of Best Practice (COBP) for Command and Control Assessment, and has the goal of establishing a common data infrastructure.[66] The C2IEDM effort has been ongoing for over 10 years, and represents a common ontology through which legacy ontologies and data models can be incorporated.

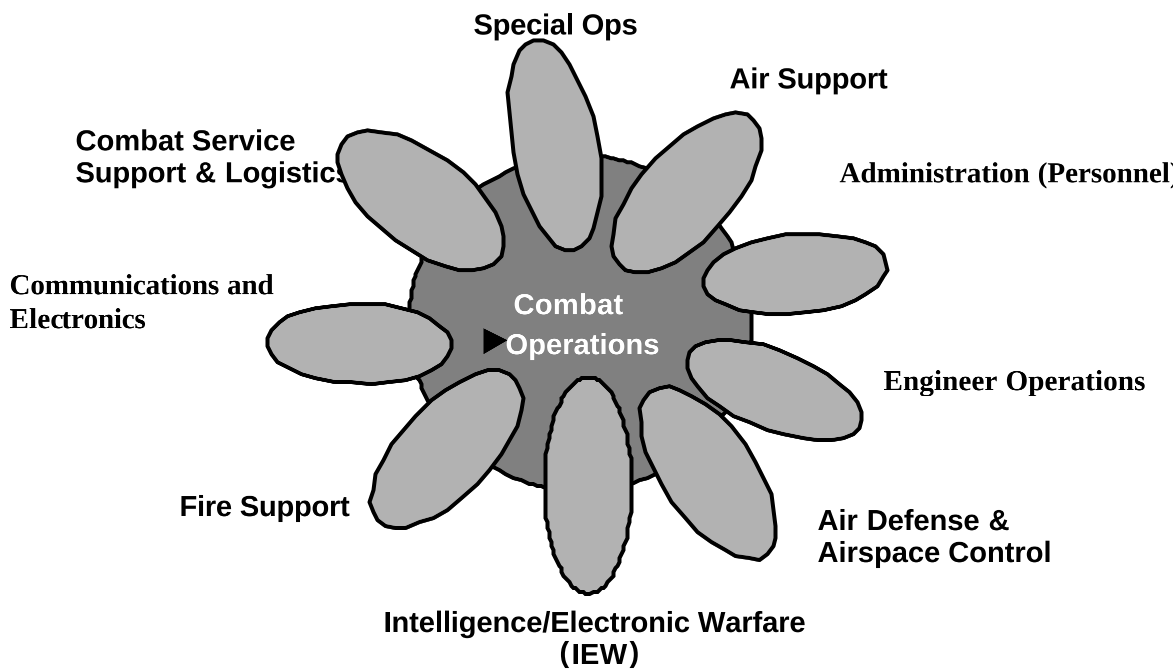


Figure 21. Generic Hub and Its Relationship to Functional Areas

In the current system of systems, separate conversions are needed to share data between any two databases. The common ontology concept proposes that each database establish the ability to convert to one central database, thereby achieving Network-Centricity and interoperability without having specific knowledge of external database structures. This is an ‘n to one’ solution, as opposed to the ‘n to n’ approach which has proven unfeasible and impractical. In this context the C2IEDM can be considered as an ‘articulation ontology,’ through which data can be published to the GIG. The C2IEDM is

---

66 Tolk, Andreas, and Sinclair, Mark R. “*Building up a Common Data Infrastructure*”, NATO Symposium on the “Analysis of the Military Effectiveness of Future C2 Concepts and Systems” organized by the Studies, Analyses and Simulation Panel (SAS), NC3 Agency, The Hague, The Netherlands, April 2002.

recognized as a well established and complete data model that can accommodate a high percentage of all command and control information exchange needs[67]. If the C2IEDM is implemented in an extensible manner it represents an ideal starting point upon which to build and adjust a data model that will accommodate the vast and ever-changing needs of the warfighter.

Because of its completeness, and comprehensive structure, the C2IEDM is an ideal model for an XML based ontology that brings the power of XML validation and XML transformation to the C2IEDM, and allows application and database independent implementation of the n-to-1 information exchange model.

C. DATABASE REPLICATION: THE ATTCIS REPLICATION MECHANISM (ARM)

The ATTCIS Replication Mechanism (ARM) is an existing database schema that instantiates the C2IEDM as a common database that organizations can use to facilitate information exchange. All units that maintain ARM databases can achieve interoperability and information exchange at the database level. The ARM applies traditional database methodologies to allow data level information exchange. The challenge for developers is to understand and apply the C2IEDM in software. The exemplar in this chapter exposes this database as an XML Schema. The XML Schema representation of the ARM database schema is automatically generated using XSLT, and demonstrates the power of XML technology.

D. EXEMPLAR: EXPRESSION OF THE C2IEDM USING XML SCHEMA

1. Introduction

There is great potential for the use of the XML, XML Schema, and the XSLT in the development of structured data in the context of Network-Centric warfare.[68] Because Network-Centric warfare will rely on a federated database model vice a centralized database system[69], it is important to develop a

---

[67] Ibid 64

[68] Ibid. 5

[69] Tolk, Andreas, "*Bridging the Data Gap – Recommendations for Short, Medium, and Long Term Solutions*", Paper 01S-SIW-011 at the Spring Simulation Interoperability Workshop 2001, Orlando, Florida, March 2001

capability to expose existing database models so that their content can be made available on the GIG without the requirement of proprietary database application interfaces.

XML Schema was developed to implement strict data typing, tree-based hierarchical relationships, domain integrity, and validation using XML. Because XML Schemas are XML documents themselves, they can be generated using XSLT. XML Schemas can be used to encapsulate the logical and semantic information that is used to define databases. Traditional table-based relational databases can be expressed as tree-based XML data structures by expressing the database schema in terms of an XML Schema. This work describes an automated process that uses a relational database schema as documented in tables to generate a corresponding XML Schema.

2. Database Schema Description

The new concept of federated databases requires that legacy systems be merged into the new system of systems without having to change the legacy system itself.[70] To the extent that databases are well defined, this can be accomplished using XML, XML Schema, and XSLT. If database schemas are not well documented, then a process must be undertaken to describe them in a format that logically describes their relationships, datatypes, attributes and entities so that they can be expressed in XML. Often it is possible to transcribe this metadata directly into XML documents, but in the case of large and complex databases it is often more practical to create tables for the meta-data information. The C2IEDM takes this approach and provides these tables in its documentation.[71] This work uses these tables to automate the creation of an XML Schema that directly models the database schema of the C2IEDM.

3. Why XML Schema?

It is perfectly feasible to simply use another database in which to maintain the metadata that comprises a database schema. This can even be used as an

[70] Ibid. 66

[71] NATO, Army Tactical Command and Control Information System (ATCCIS) Working Group, *The Land C2 Information Exchange Data Model, Working Paper 5-5, Edition 5.0, ATCCIS Baseline 2.0 - ANNEXES*, 18 March 2002

intermediate step towards the development of an XML Schema, but it lacks some important characteristics that make XML Schema a powerful device for Network-Centricity. A database schema that is expressed as an XML Schema takes the form of one or several text documents. XML is platform independent, application independent, and can be distributed by existing internet systems. A database, on the other hand, is application and platform dependent, and requires specific interfaces for web delivery. Database applications prevent the complete separation of data and presentation and impose proprietary interface requirements that are prohibitive to interoperability.

XML Schema and XSLT are Network-Centric tools because of the ubiquitous availability of validating parsers and stylesheet transformation engines on the network. These basic utility programs are built into current web browsers and can be incorporated as standalone devices, or into interface software in order to provide transformation and validation functionality to any system. XSLT is a Turing complete programming language[72] that can recursively analyze, compare, and transform XML defined data from one format to another. Like XML Schema, an XML Stylesheet can be delivered as a text document, and can be associated directly with an XML instance document in such a way as to automate validation and transformation processes.

The most important function of an XML Schema is that it can be used for validation. This is a process by which a widely available and standardized software utility, a validating parser, can verify that a particular instance of XML is an instance of a particular XML Schema. Validating XML parsers are integrated into all current web browsers and will be standard issue for all Network-Centric applications. An XML document that is validated as an instance of an XML Schema is compliant to the requirements for data types, relationships, entities, attributes, and context; and can be considered as equivalent to a traditional database entry. While a traditional database schema is principally created for human consumption in order to describe functionality and requirements, XML

[72] Kay, Michael, *XSLT Programmer's Reference 2nd Edition*, Wrox Press, 2001

Schema is dual purposed for machine processing so that data validity can be assured before it is published for use by applications. One of the most common subscribers to validated data will be databases themselves.

4. Articulation Ontology

The XML version of the C2IEDM is not meant to be used directly by applications, but is rather as a common target for transformations. The reason for this is that the C2IEDM must necessarily change over time. The loose coupling that the transformation mechanism provides will allow applications to accommodate this change by adjusting the intermediate transformations, rather than by redesign and re-development of software. For this reason, a central ontology like the one created from the C2IEDM is considered to be an "Articulation Ontology,"[73] through which information exchange is accomplished. In order to formulate the C2IEDM as an articulation ontology it is first necessary to express it in the form of an XML Schema so that external data models will have a target for which to develop XSLT transformations.

5. Database Schema vs. XML Schema

It is important not to confuse the tables that document the database schema with the tables that comprise the relational database. The database schema tables simply illustrate how data in the actual tables is organized and related. The term replication mechanism applies to the way that a C2IEDM relational database creates, or replicates an instance, in order to store and exchange data with other ATCCIS defined C2IEDM databases. The specification states that "when a Command and Control(C2) application changes the state of information that it holds, and which is recognized by the ATCCIS specification, this information is automatically replicated to all other co-operating systems that have agreed to exchange this information. The meaning and context of the information is preserved and requires no additional processing on receipt to make it useful."[74]

[73] Kogut, Paul, Cranefield, Stephen, Hart, Lewis, Dutra, Mark, Baclawski, Kenneth, Kokar, Mieczyslaw and Smith, Jeffrey, *UML for Ontology Development*, Knowledge Engineering Review Journal Special Issue on Ontologies in Agent Systems, Vol. 17, 2002

[74] Ibid. 64

The Common Operating Environment (COE) is evolving into a new architecture called Net Centric Enterprise Services (NCES) which will provide publish/subscribe services that allow warfighters to pull or submit any information to and from any available network sources, at any time.[75] The C2IEDM clearly addresses this requirement, but must be extended so that it can maintain the advantages that the relational database core provides. An XML Schema that exactly models the ATCCIS Replication Mechanism is useful because it allows entry into the C2IEDM system on the data level. XML Schema provides the ability to maintain the context and meaning of data with respect to the C2IEDM by providing structure and validation. Of course this approach is not limited to the ATTCIS Replication Mechanism. It can be applied to almost any distributed database system. An automated conversion from database schema to XML Schema facilitates the process of making powerful databases accessible to the NCES.

A XML savvy reader is surely aware that such capabilities are already built in to many XML authoring tools and major database platforms. These tools serve to support the idea that this is a needed and useful function, but they are not yet capable of accurately reflecting a written database schema. The functionality of existing automated tools is limited to the creation of an XML Schema based on an instance representation of a relational database. This can be a partial solution, but there is still a great deal of manual effort required to enter all of the correct datatypes, to ensure that the relationships are correctly modeled in the tree structure, and to fully annotate the XML Schema. Usually XML tools and database engines produce XML expressions of table data structures. These are useful, but they are a far cry from a functional XML Schema that can be used to produce valid instances of a relational database.

[75] Tolk, Andreas, "A Common Framework for Military M&S and C4I Systems", 2003 Spring Simulation Interoperability Workshop, April 2003

6. The ATCCIS Replication Mechanism Schema

An important initial step in this process was the development of an XML Schema by Francisco Laoiza, at the Institute for Defense Analysis (IDA) that faithfully models the ATCCIS replication mechanism.[76] The creation of this XML Schema was partially automated using an XML authoring tool and partially adjusted manually. This version models the ATCCIS Replication Mechanism directly and is referred to as the Battlespace Generic Hub - ATTCIS Replication Model (BGH-ARM) Schema.

The BGH-ARM Schema is far smaller and less complicated than the one generated in this exemplar because it does not encapsulate enumeration values or represent database relationships in an extended tree structure. It also does not include many of the annotations that are helpful for reconciling data from external sources with the data model. The BGH-ARM is an important tool in the process of exposing the C2IEDM to the NCES because it faithfully represents the database incarnation of the C2IEDM, but it does not fully implement the advantages of XML Schema.

The fact that there can be several different XML Schemas that represent a single data model is a key benefit of extensible technologies. The C2IEDM information exchange mechanism is expressed as a relational database, but was created as an object oriented database that can accommodate functionality beyond the ATCCIS Replication Mechanism. An XML Schema that represents the C2IEDM with more detail can be used as a vehicle through which to transform to the BGH-ARM Schema, as well as a reference mechanism that can be used to create an XML language based on the C2IEDM.

7. Source Document Correlation

All of the entities, relationships, cardinalities, enumerations, key types, and identifiers that make up the C2IEDM are contained in very large tables in the specification. The task is to generate a comprehensive XML Schema that explicitly contains all of the information contained in the specification. The intent

[76] BGH-ARM Schema, or GH5Complete.xsd included in supplemental code package.

is to show that XML Schema can be derived from document based sources and that a direct correlation between document based information and XML based information can be established. If the XML Schema can be auto generated from the tables that describe the C2IEDM BGH data model, then future changes in the specification can be reflected in the schema simply by repeating the auto generation process. If there are changes to the format or arrangement of the tables in the specification, the only adjustment required is in the XSLT stylesheets.

In order to perform the auto generation it is necessary to convert the C2IEDM document format from MSWord to OpenOffice.org. This file format conversion maintains all formatting, but the OpenOffice.org document has the advantage of having a native XML format. Because the OpenOffice.org format is based on an open source schema that has been developed to represent standard office suite data, it is possible to use XSLT to extract the data from the XML described embedded tables into raw XML documents. After this is done, another XSLT script is applied to combine these tables in accordance with the relationships that the information in them specifies. In essence, the open office data structures containing information that describe the C2IEDM data structure are transformed into an XML Schema that represents the C2IEDM.

The auto generation of the XML Schema from source documentation serves as an example of the way that a great number of authoritative documents that govern battlespace information generation and exchange can be directly associated with XML Schema so that changes will be automated. The concepts of human readability and machine readability are extended here in that the same data models are being viewed in completely different, but compatible documents. The text based human readable document is reflected in the XML Schema by virtue of the transformation mechanism that XSLT provides.

8. Auto Generating an XML Schema from Text Document Tables

Because the tables in the C2IEDM are so large it is very difficult to perform a conversion manually. The need to automate the process is also a matter of conceptual limitations. The entities and relationships described in the

C2IEDM are database specific and involve many repetitive and complex structures. Manual interpretation of these structures are prone to human error and to incorrect interpretation. Automation ensures a consistent and exact duplication of the relationships described.

Figure 22 shows the top level entities in the C2IEDM. These entities have many more explicit relationships that are not shown in the diagram. The diagram illustrates a many to many relationship between all entities. This creates a extremely complex data structure when it is instantiated in a tabular database.

Figure 23 and 24 describe a small portion of the ATCCIS data model. Figure 25 illustrates the data structure in a relational database environment. Figure 26 is a graphical representation of the OBJECT ITEM table in the resulting XML Schema, and Figure 27 shows the textual content in XML. Figure 28 depicts the entire procedure that is followed to create a schema that expresses all of the relationships in the Database Schema. Each table was processed into a basic XML representation, and then combined according to content in order to create the final schema.

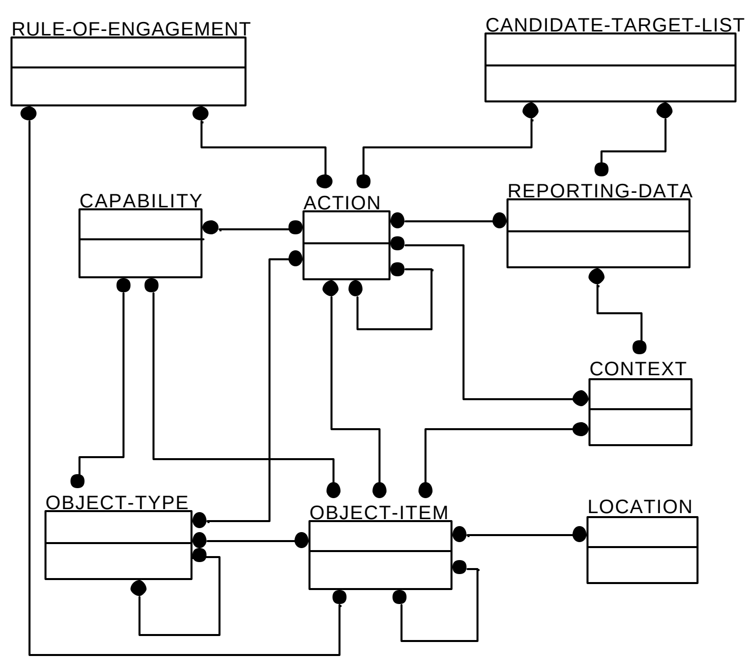


Figure 22. Key Entities of the Generic Hub Data Model

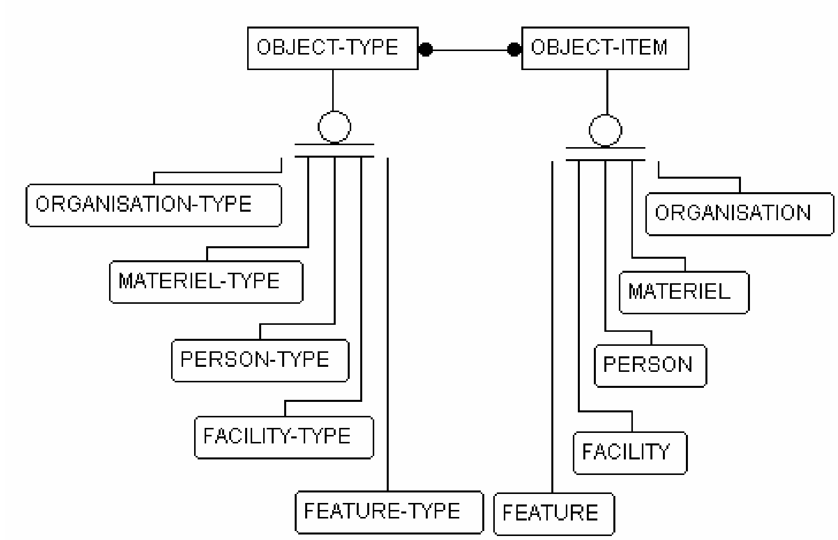


Figure 23. IDEFX Diagram describing a relation

Entity	Entity Definition
FACILITY	An OBJECT-ITEM that is built, installed, or established to serve some particular purpose and is identified by the service it provides rather than by its content.
FACILITY-TYPE	An OBJECT-TYPE that is intended to be built, installed or established to serve some particular purpose and is identified by the service it is intended to provide rather than by its content. Examples include a refueling point, a field hospital, a command post.
FEATURE	An OBJECT-ITEM that encompasses meteorological, geographic, and control features of military significance.
FEATURE-TYPE	An OBJECT-TYPE that encompasses meteorological, geographic, and control features of military significance. Examples include a forest, an area of rain, a river, an area of responsibility.
MATERIEL	An OBJECT-ITEM that is equipment, apparatus or supplies without distinction as to its application for administrative or combat purposes.
MATERIEL-TYPE	An OBJECT-TYPE that represents equipment, apparatus or supplies of military interest without distinction to its application for administrative or combat purposes. Examples include ships, tanks, self-propelled weapons, aircraft, etc., and related spares, repair parts, and support equipment, but excluding real property, installations, and utilities.
ORGANISATION	An OBJECT-ITEM that is an administrative or functional structure.
ORGANISATION-TYPE	An OBJECT-TYPE that represents administrative or functional structures.
PERSON	An OBJECT-ITEM that is a human being to whom military significance is attached.
PERSON-TYPE	An OBJECT-TYPE that represents human beings about whom information is to be held.

Figure 24. Definition of First-Level Subtypes

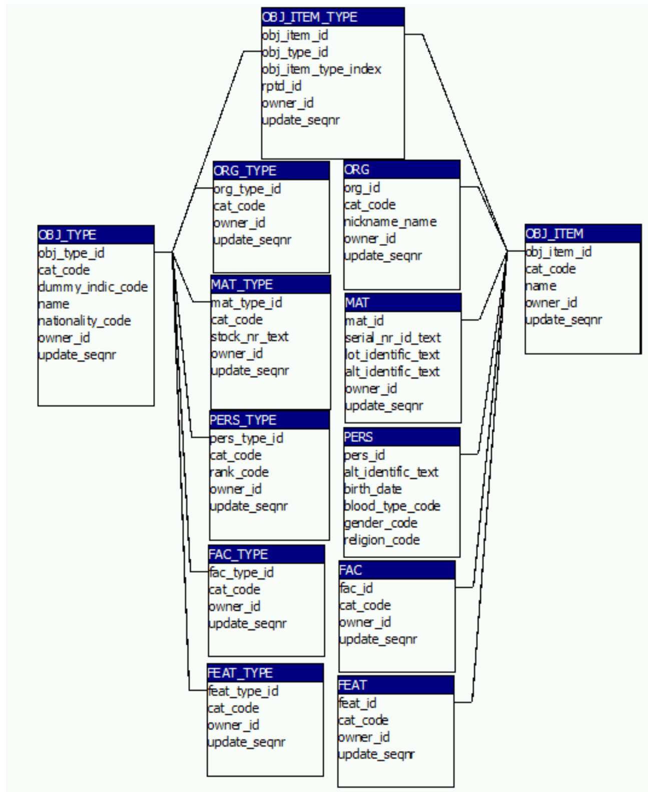


Figure 25. Tabular Database Structure

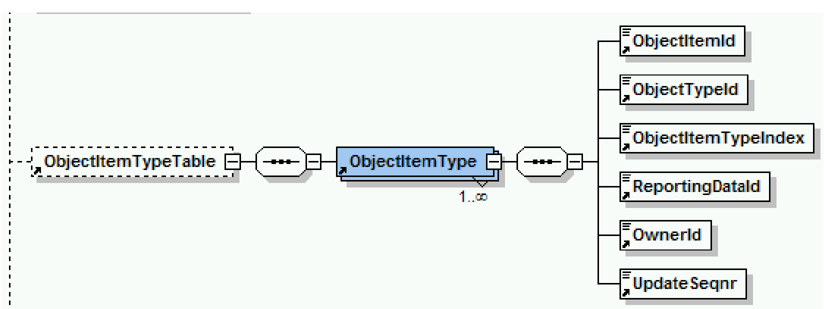


Figure 26. OBJECT-ITEM Representation

```
<xs:element name="ObjectItemTable">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ObjectItem" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  </xs:element>
  <xs:element name="ObjectItemType">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ObjectItemId"/>
        <xs:element ref="ObjectTypeId"/>
        <xs:element ref="ObjectItemTypeIndex"/>
```

Figure 27. XML Schema Segment

9. Database Schema Conversion

The C2IEDM is a very complex data model that is not very well suited for manual interpretation and conversion into XML. The tables that describe the C2IEDM are contained in a set of Annexes that use an extended database entity-relational model called IDEF1X.[77] The tables contain the IDEF1X data model diagram, entity definitions and attributes, entity relationships, attribute definitions, specifications for enumerated domains, and business rules.

XML Schema and XSLT documents are actually physical representations of a process. It is hoped that this exemplar will provide a reference for future maintenance of this and other database projects, so that the appropriate expertise is applied. It is be apparent that this methodology can extend beyond databases to include almost any structured, or semi-structured data products.

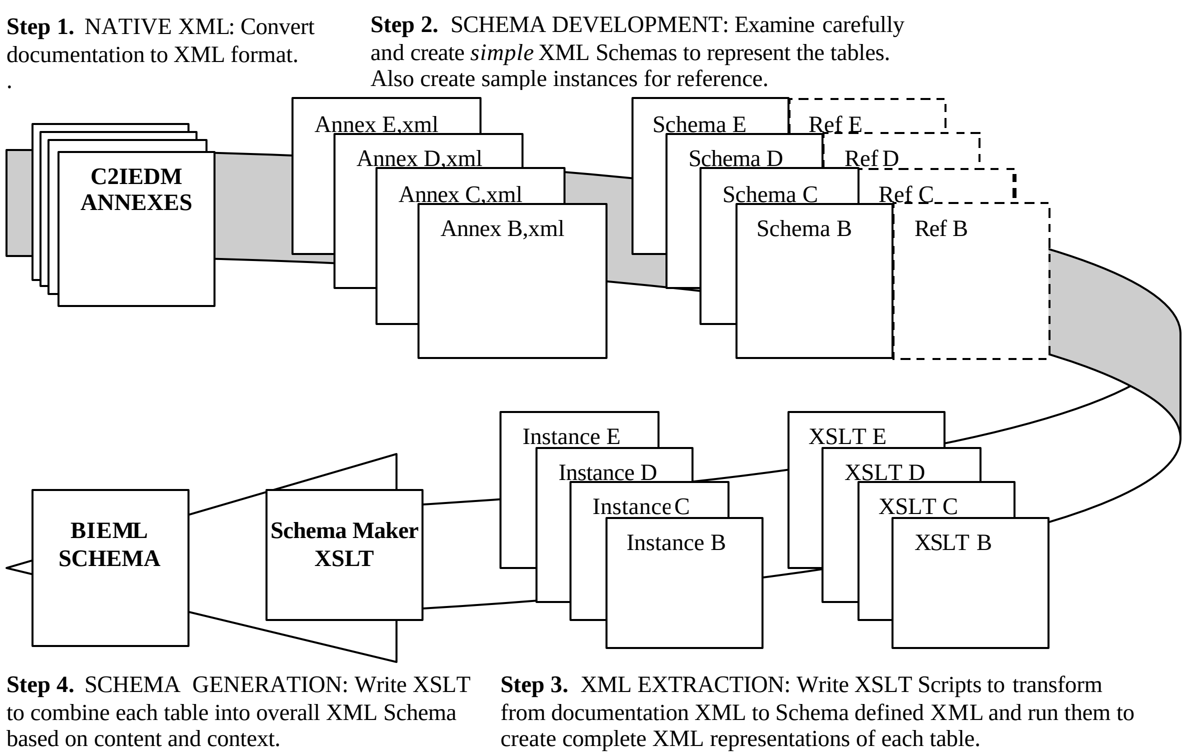


Figure 28. Creating an XML Schema From Source Documentation

---

[77] NATO, Army Tactical Command and Control Information System (ATCCIS) Working Group, "*Annexes A-K: Data Model Documentation,*" Working Paper 5-5, Edition 5.0, ATCCIS Baseline 2.0, 18 March 2002

Step 1: Convert Text Document Content to Native XML

The first step in creating an XML Schema from Database Schema documentation is to render that documentation in a native XML format. This was done using OpenOffice.org[78], which is dedicated to the purpose of making standard office type data available in open source, native XML formats. In an effort to assist forward thinking enterprises, this open source effort has included the ability to convert MSOffice suite documents to the OpenOfffce.org native XML format. Both the MSOffice and OpenOffice.org versions of the C2IEDM Documentation and Annexes are made available in the supplemental code package that is required for full understanding of this work.

An important takeaway from the first step is to realize that native XML formats for all technical specifications will provide the advantage of machine readability without the need for prior manual conversions. This might obviate the first step and simplify some of the XSLT operations in subsequent steps. Because of the unstructured nature of Office based tables, the XSLT operations must be customized for each table. This may seem overly difficult at first, but given the power and flexibility that is at hand it is apparent that this methodology is far easier to maintain than one which relies on byte code.

Step 2: Create XML Schemas to Model Tables

The creation of naming standards, global element requirements, and design parameters is identified by Rosenthal et al. as a process of "choosing a formalism for community information models." These authors recognize that with the abundance of transformation tools available there can be many different and simultaneous formalisms that each depend on different purposes[79]. These XML Schemas are used for the conversion of the C2IEDM database schema tables into XML Schema, but they can be regarded as an entry point for the expression of any database schema using XML Schema.

[78] www.openoffice.org, Accessed 08/27/2003

[79] Rosenthal, Arnon, Seligman, Len and Costello, Roger, "XML, Databases, and Interoperability", Federal Database Colloquium, AFCEA, San Diego, 1999

XML Schemas for XML instances that are to be used specifically as the source data for XSLT transformations must be highly accessible, so the best rule of thumb to follow is simplicity. XSLT applies the XPath[80] query language and can accommodate a great deal of complexity in XML structure and Schema design. The challenge, however, is to make the Schemas understandable and useful to human developers so that the implementation can be replicated and extended for other uses, and will not have to be re-written from scratch.

Figures 29 through 32 describe Step 2 in detail. They show partial views of the text based tables of Entity Definitions and Attributes from Annex B, Entity Relationships from Annex C, Attribute Definitions from Annex D, and Enumerated Domains form Annex E. Each illustration shows an XML Schema diagram of the kind that is commonly used in XML authoring tools.[81] Also depicted are the text versions of the XML Schemas and XML instances that are annotated to indicate the type of data fill that is expected. The XML instance representations are the target output for the XSLT scripts that were created to extract the data from the C2IEDM documentation. All of the XML Schemas, and XSLT scripts are provided in Appendix D. The resulting instances are too large to print and are provided in the code package.

The validation rules that are contained in Annex J of the specification are not included in this treatment. For future work, this data can be added as a separate table, and referenced during the schema auto-generation process in order to add the limits and data typing that are described there. Similarly, Annex I contains the Structured Query Language (SQL) commands that are required to instantiate the tables in the documentation. These too can be incorporated into the Schema, either as annotations or in text elements, and can be used to complete the cycle between the XML Schema and the database schema.

There are undoubtedly other concerns and features that can be addressed using this methodology. There are also design factors that can be implemented

[80] World Wide Web Consortium, W3C Recommendation, “XML Path Language (XPath) Version 1.0”, [http://www.w3.org/TR/2003/WD-xpath20-20030502]. November 1999

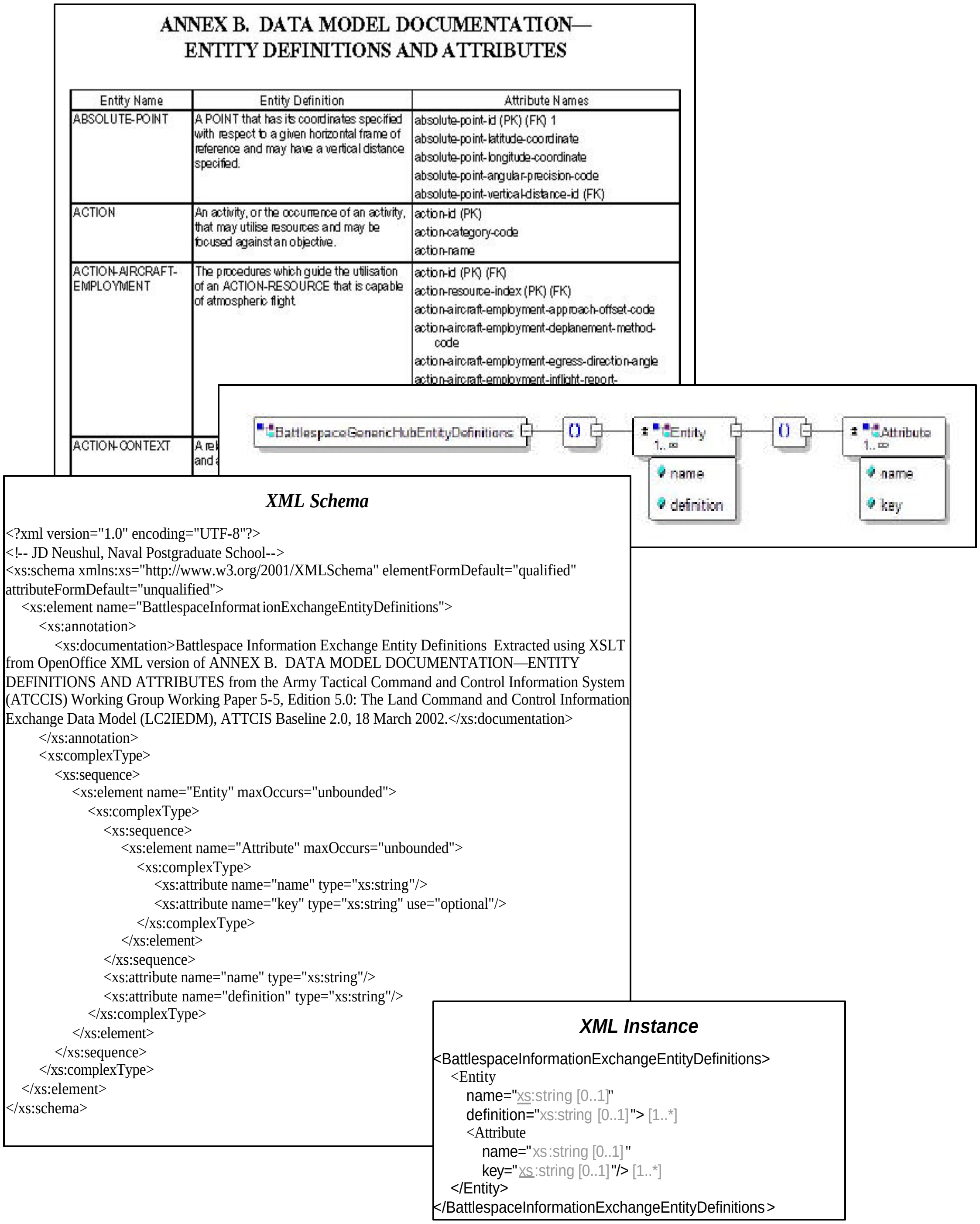
[81] Diagrams generated by Stylus Studio from Sonic Software Corporation.

in the documentation to make improvements possible. In this initial, reference implementation it was decided that exhaustive data typing and inclusion of other code might be better integrated if it were addressed more thoroughly first in the documentation. This work is meant to increase understanding and attention to this use of documentation so that design will be adjusted accordingly.

Step 3: Write XSLT to Transform OpenOffice.org XML to Schema Defined XML

OpenOffice.org files are saved as compressed collections of xml documents that contain content, settings and styles. The XML content from the Annexes document was extracted using a standard decompression tool. The original documentation, OpenOffice.org version, and extracted content are included in the supplementary code that can be used if the intent is to implement the techniques that are described in this paper.

The XSLT Stylesheets that were developed to perform these extractions are annotated extensively to provide reference on how the XSLT works and to demonstrate the various problems that are associated with transforming relatively flat-structured document style XML to a tree-structured XML. These stylesheets are provided in Appendix B for reference.



ANNEX B. DATA MODEL DOCUMENTATION— ENTITY DEFINITIONS AND ATTRIBUTES

Entity Name	Entity Definition	Attribute Names
ABSOLUTE-POINT	A POINT that has its coordinates specified with respect to a given horizontal frame of reference and may have a vertical distance specified.	absolute-point-id (PK) (FK) 1 absolute-point-latitude-coordinate absolute-point-longitude-coordinate absolute-point-angular-precision-code absolute-point-vertical-distance-id (FK)
ACTION	An activity, or the occurrence of an activity, that may utilise resources and may be focused against an objective.	action-id (PK) action-category-code action-name
ACTION-AIRCRAFT-EMPLOYMENT	The procedures which guide the utilisation of an ACTION-RESOURCE that is capable of atmospheric flight.	action-id (PK) (FK) action-resource-index (PK) (FK) action-aircraft-employment-approach-offset-code action-aircraft-employment-deplanement-method-code action-aircraft-employment-egress-direction-angle action-aircraft-employment-inflight-report-
ACTION-CONTEXT	A rel... and ...	

XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="BattlespaceInformationExchangeEntityDefinitions">
      <xs:annotation>
         <xs:documentation>Battlespace Information Exchange Entity Definitions  Extracted using XSLT
from OpenOffice XML version of ANNEX B.  DATA MODEL DOCUMENTATION—ENTITY
DEFINITIONS AND ATTRIBUTES from the Army Tactical Command and Control Information System
(ATCCIS) Working Group Working Paper 5-5, Edition 5.0: The Land Command and Control Information
Exchange Data Model (LC2IEDM), ATTCIS Baseline 2.0, 18 March 2002.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Entity" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Attribute" maxOccurs="unbounded">
                        <xs:complexType>
                           <xs:attribute name="name" type="xs:string"/>
                           <xs:attribute name="key" type="xs:string" use="optional"/>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
                  <xs:attribute name="name" type="xs:string"/>
                  <xs:attribute name="definition" type="xs:string"/>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

XML Instance

```
<BattlespaceInformationExchangeEntityDefinitions>
   <Entity
      name="xs:string [0..1]"
      definition="xs:string [0..1]"> [1..*]
      <Attribute
         name="xs:string [0..1]"
         key="xs:string [0..1]"/> [1..*]
   </Entity>
</BattlespaceInformationExchangeEntityDefinitions>
```

Figure 29. XML Schema Diagram Representation, XML Schema and XML Instance of Entity Definitions and Attributes Table of the C2IEDM Annexes.

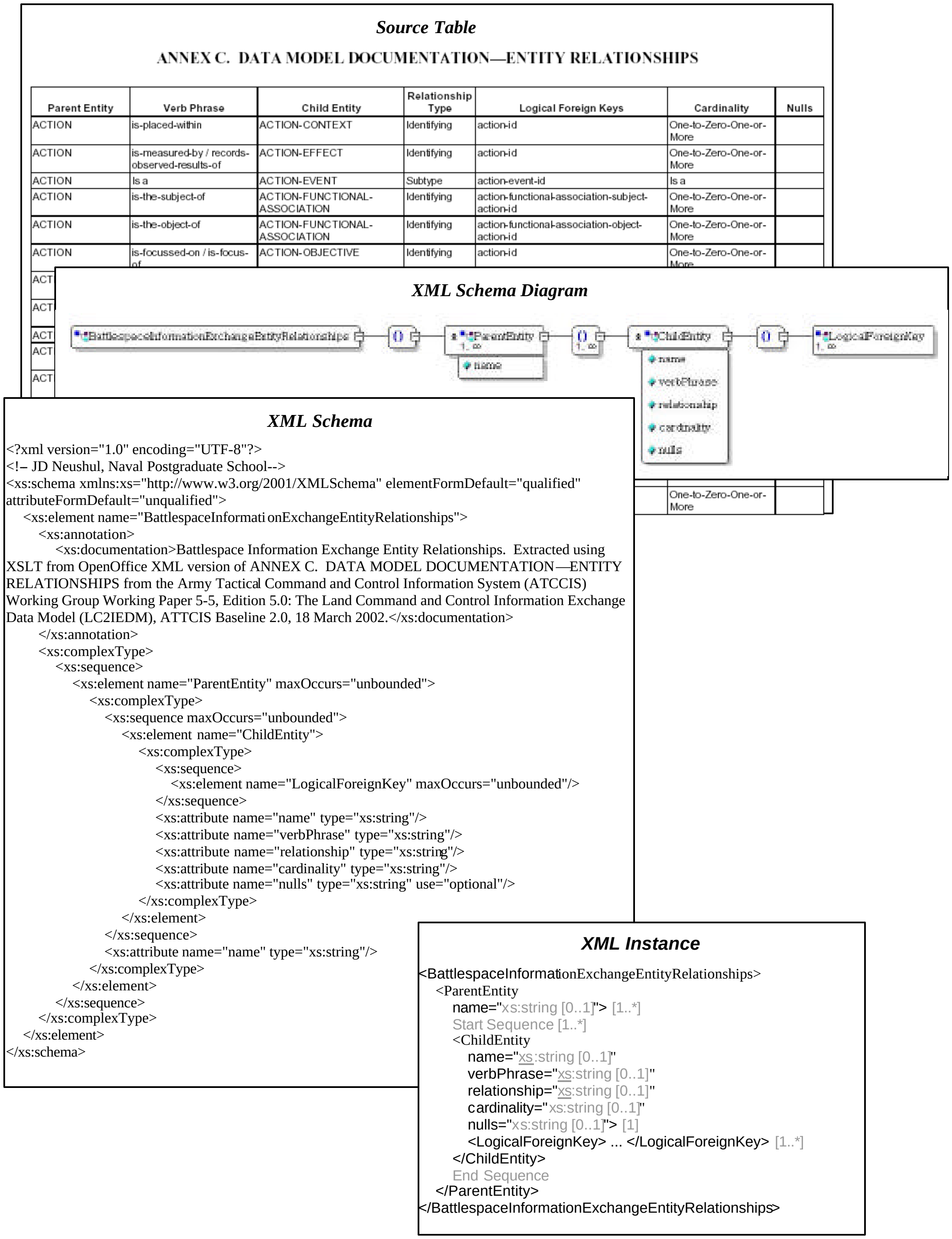


Figure 30. XML Schema Diagram Representation, XML Schema and XML Instance of Entity Relationships Table of the C2IEDM Annexes.

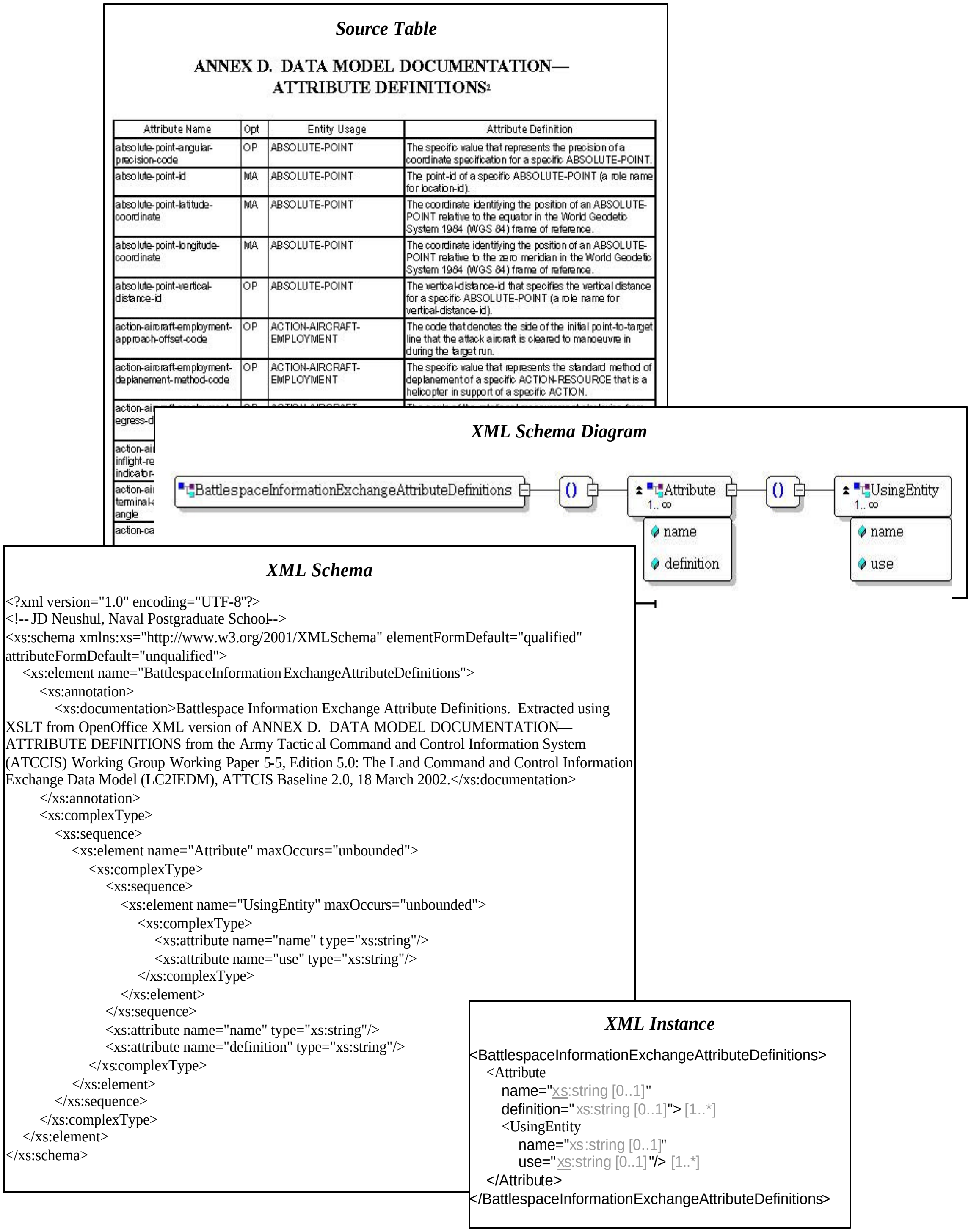


Figure 31. XML Schema Diagram Representation, XML Schema and XML Instance of Attribute Definitions Table of the C2IEDM Annexes.

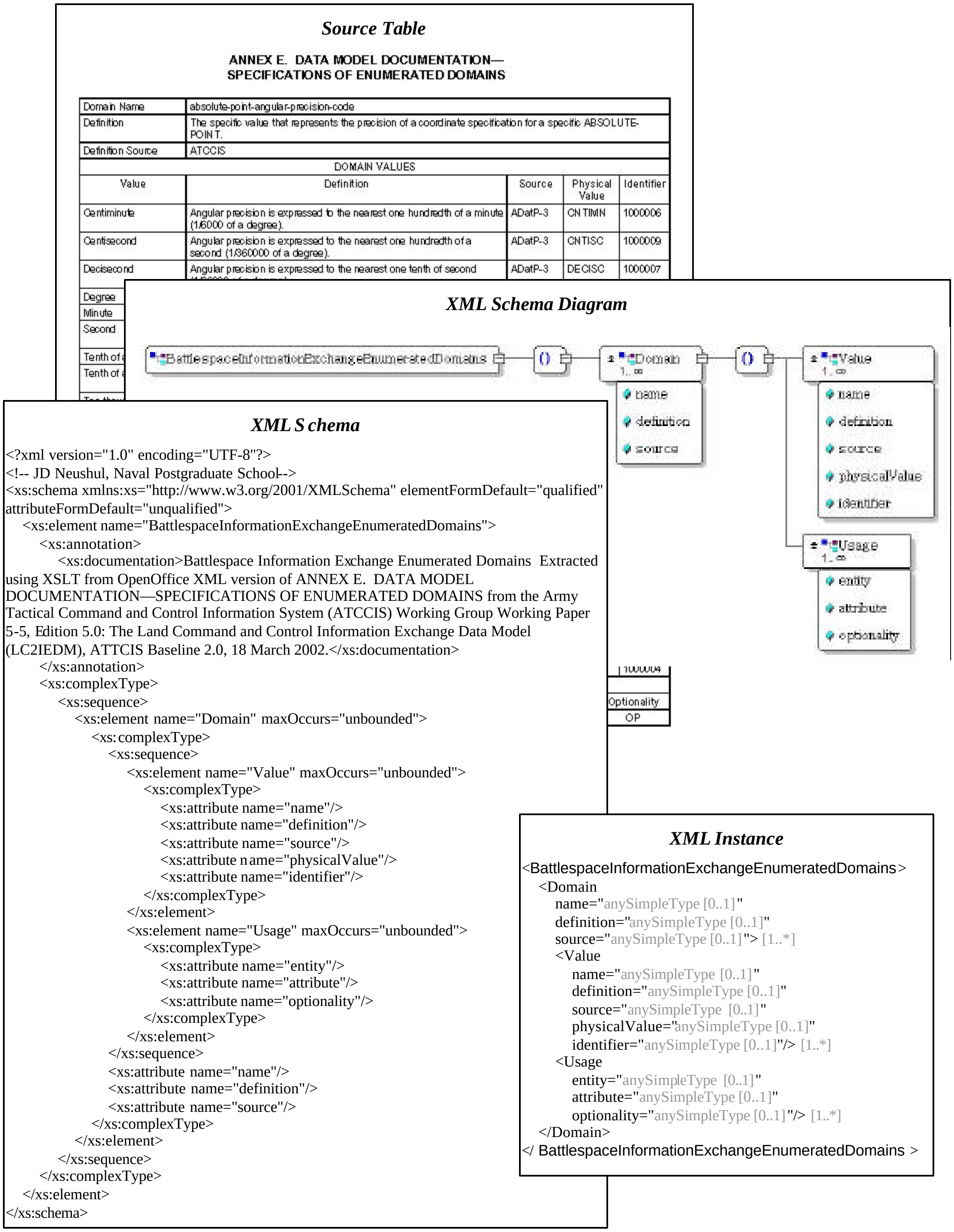


Figure 32. XML Schema Diagram Representation, XML Schema and XML Instance of Enumerated Domains Table of the C2IEDM Annexes.

The XSLT in these examples has been tested with the Apache XALAN[82] XSLT Parser, and the built in parser to the XML authoring application Stylus Studio, by Sonic Software[83]. It worked with some MSXML[84] parsers, and not with others. There is often some disparity between XSLT Parsers, usually instigated by proprietary interests. For this reason the MSXML Parser was avoided. There are specific methods that XSLT provides to ameliorate differences, just as the HTML includes ways to accommodate differences in browsers. This is a symptom of unreliability and inconvenience in emergent XML technology similar to that in web browsers. This may detract from the web browser as a success model, but it cannot be denied that, despite the inconveniences, web browsers have still managed to become the most ubiquitous Network-Centric tools on the planet. XML is designed to follow this well beaten path.

Step 4: Write XSLT That Uses the Data in the XML Tables to Create an XML Schema

The final XSLT Program in this process is the most significant for general purpose conversion from database schema to XML Schema. In this script all of the relationships that are described in the database schema are analyzed and converted to a tree based format. The tree based data structure that is inherent to XML is the most pertinent difference between XML data structures and traditional table based database structures. The conversion of table databases to tree structure introduces the powerful concepts of scope and context.

The creation of global elements in XML Schema serves to prevent repetition and to identify the building blocks of the data. In this case the use of global elements makes the resulting XML Schema a far more manageable size than versions that did not create global elements. Early iterations that did not use global elements were 3 to 4 times the size and became difficult to manipulate and analyze. The size of the final Battlespace Information Exchange Schema is

[82] Apache Software Foundation, www.apache.org, Accessed September 2003

[83] Sonic Software Corporation, www.sonic.com, Accessed September 2003

[84] Microsoft Corporation, www.microsoft.com, Accessed September 2003

still very large and will offend the sensibilities of some XML designers. Beyond assigning global elements, a remedy for the size might be to create several XML Schemas and have a central one reference those. The best way of doing this might be to create a separate schema for each of the key entities illustrated in Figure 33. This diagram displays the many-to-many relationships between these key entities, so each schema in a multiple schema model might have to reference every other schema. This may be a practical approach for future work, but for the purposes of simplicity and clarity the single schema approach is taken here.

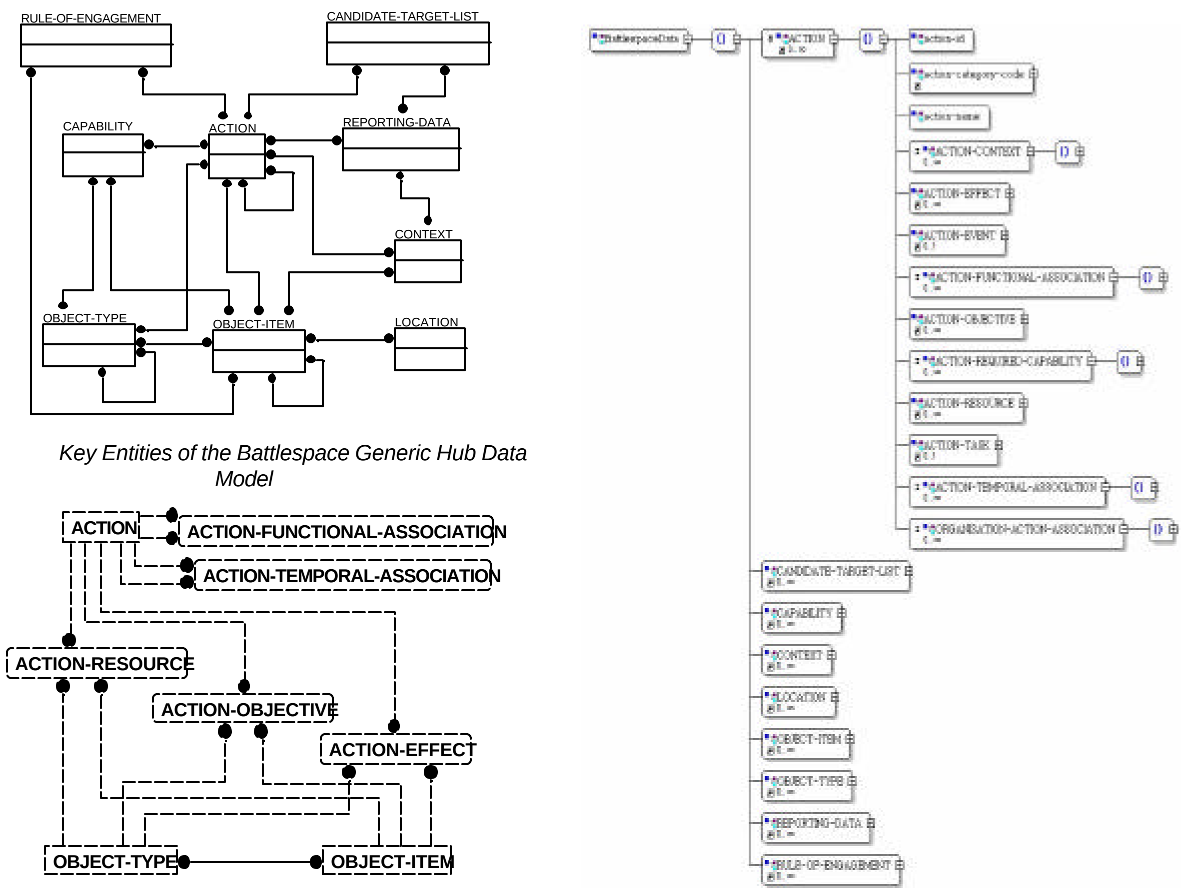


Figure 33. Key Entities of the battlespace Generic Hub Data Model, and The Parent Elements of the Auto-generated battlespace Information Exchange Mark-up Language (BIEML).

Figure 33 shows a diagram of the top level parent entities that are designated using the MakeSchema.xsd. The choice of top level entities is driven by the documentation. Analysis of the details of the database schema is necessary to establish the basic patterns to follow in its interpretation. The relationships that are indicated by the Key Entity diagram are represented in the tree structure of the BIEML diagram. The first level of the ACTION element shows that in addition to many of the database-centric data elements, the other key entities are referenced.

The MakeSchema.xsd XSLT is a very powerful XSLT program because it allows the designation of a few basic patterns and then performs an extremely complex and exhaustive conversion which is prohibitively difficult to perform manually. A measure of accuracy and correctness with regard to the original database schema can be found by comparing various levels of the resulting XML tree structure with the database entity-relationship diagrams that are provided in the documentation. The full Schema is too large to be printed, and is made available in the supporting code.

10. The Battlespace Information Exchange Markup Language

The expression of the C2IEDM as a fully expanded and verbose XML Schema essentially amounts to the creation of an XML defined language. The faithful representation of all of the relationships that are described in the C2IEDM is meant to facilitate the adaptation of tactical and operational data for use in the C2IEDM. The resulting Schema is intended be used in its entirety, or in part, to create and validate XML instances that can be transformed to populate an ATCCIS database. The transformation of the C2IEDM into a markup language is appropriately named by replacing the words Command and Control with "Battlespace," and the words "Data Model" with "Markup Language" to become the Battlespace Information Exchange Markup Language (BIEML). It is important to recognize that this work is a part of a larger effort that may well see fit to change, re-design, and rename this language.

Because the BIEML is simply an exact restatement of the C2IEDM it is the full and complete property of the governing body that has cognizance over the C2IEDM. Currently, this is the Multilateral Interoperability Programme (MIP), whose aim is to "achieve international interoperability of Command and Control Information Systems (C2IS) at all levels from corps to battalion, or lowest appropriate level, in order to support multinational, combined and joint operations and the advancement of digitization in the international arena including NATO."[85] This work is not officially affiliated with the MIP, but it is conducted in the spirit of this goal, with the added point that interoperability must be addressed in the same way within and between our own services.

E. APPLYING THE BATTLESPACE DATA MODEL

The Joint Staff Perspective of the MIP refers to the definition of interoperability as the "Ability of systems, units, or forces to provide services to or accept services from other systems, units, or forces and to use the services so exchanged to operate effectively together."[86] It also indicates that "Standardization enables Interoperability but it alone does not achieve the objective."[87] The in depth exploration of the data model that is required to accomplish the transformation to XML Schema makes this apparent. XML Schema addresses several key issues that are identified by the MIP with regard to Information Management, Information Topology, and procedural rules to enable technology, as well as the SECDEF goal of providing an "all source picture of the battlespace containing actionable, decision quality information through a fusion of existing databases." [88]

Figure 34 illustrates the flow of information between databases and software systems. This is a broad brush representation of how an Operations Order can be stored in databases, and referenced by software. Figure 35 is taken in part from the referenced MIP brief but the blocks with dotted lines and arrows are added to indicate the places where the use of XML and XSLT can be implemented.

---

[85] Ibid. 65

[86] Chairman of the Joint Chiefs of Staff, *Instruction (CJCSI 6212.01B), Interoperability and Supportability of National Security Systems, and Information Technology Systems*, 8 May 2000

[87] Multilateral Interoperability Programme, *Brief to 2003 DoD Standardization Symposium, LtCol Scott Hoffman, USMC, Joint Staff J6I*, 4 March 2003

[88] Ibid. 86

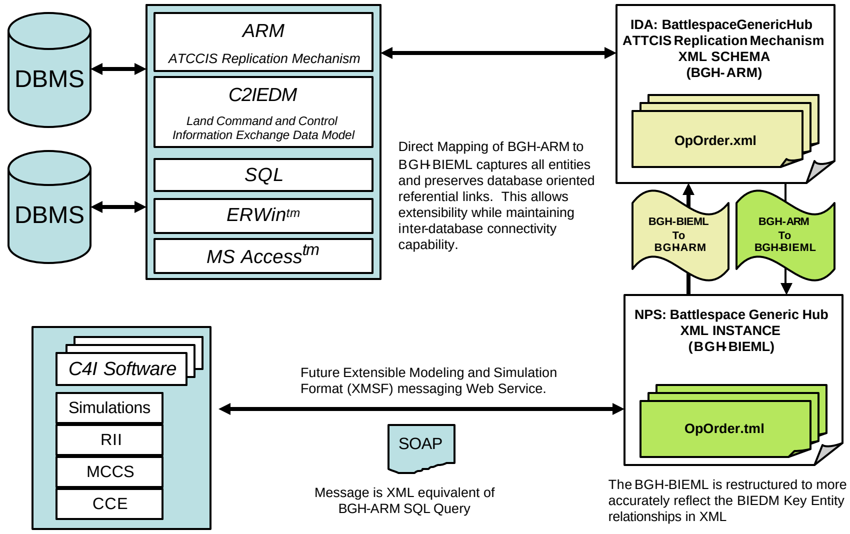


Figure 34. Information Exchange Overview

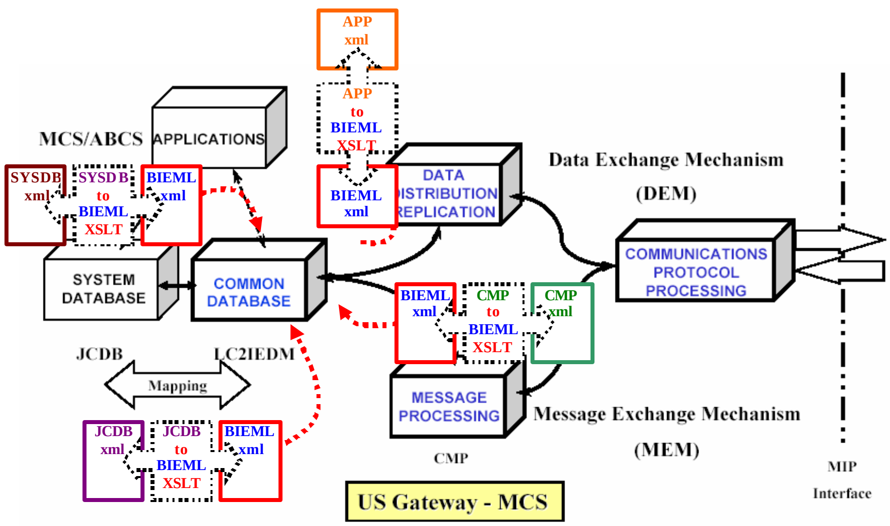


Figure 35. US Army Implementation of the C2IEDM with added XML and XSLT Extensions.

F. CONCLUSIONS

The battlespace Information Exchange Mark-up Language (BIEML) is simply a faithful reproduction of existing database efforts. The significance of this accomplishment depends on the extent to which the XML Schema can be used as a tool to connect the ATCCIS Database to databases and systems on the NCES. The XML Schema implementation has not been exhaustively tested, nor significantly implemented. Most adjustments can be made using the XSLT coupling mechanisms that have been developed.

Future work will include the implementation of all or parts of this methodology to incorporate written knowledge into structured data. A procedure to re-create the database schema tables from the XML Schema is needed and can be accomplished using XSLT. Similar methods can be used to convert between XML representations of orders and reports. The key to this approach is that it is accomplished on the data level and minimizes application dependence.

XML Schemas and XSLT Transformations are key control measures that will have to be developed in all military functional areas. The recognition of these tools as control measures is an important leadership mandate that will ensure that the warfighter maintains control over institutional ontologies, priorities, orders, directives and doctrine. There is a real danger of giving proprietary software tools de-facto control over the communication processes which are the sole responsibility of leadership.

This example demonstrates that an extremely complex database can be automatically expressed using XML Schema. The intent is to illustrate that this process can be repeated to achieve the same results with data structures that are far less complex. Many forms of military communication have evolved because they are simple, direct, efficient, and because they support the mission oriented ethos of top down oversight and bottom up execution. In this vein it is hoped that a situation will prevail in which operators and military professionals take direct responsibility for the data structures that define their activities. Because source documentation can be used to accomplish this, it is apparent that the means to achieve data control are available and accessible.

Most specifications, orders, doctrinal publications, maintenance manuals, and message formats can be expressed in XML Schema so that they can become the basis of organizational ontologies to which software must conform. Early adoption of the available tools demonstrated here will reduce the price of entry, and hasten the advancement of the new paradigm.

VII. EXTENSIBLE AUTONOMOUS AGENTS

A. INTRODUCTION

Autonomous vehicles of all types are populating the battlefield in increasing numbers and roles. Interoperability and compatibility present important design considerations, while effective employment is a command responsibility. Autonomous vehicles contribute significantly to battlespace visualization capabilities, but also present difficult command and control complexities.

This chapter outlines a methodology by which to mitigate the complexity that is introduced by the presence of multiple, and disparate autonomous systems on the battlefield. Data Control is achieved by using XML Schemas to define the behavioral factors that govern the behavior of Unmanned Autonomous Aerial Vehicles (UAAVs). The principles of multi-agent systems and the cooperative emergent behavior that is produced reduce the number of parameters that commanders and operators must provide in order to maximize the potential of multiple UAAVs. The expression of UAAV characteristics and environment using XML also provides control measures by which autonomous vehicles of different design and manufacture can be integrated in a common community.

B. OVERVIEW

This chapter asserts that interoperability and command responsibilities are not mutually exclusive, and that command prerogatives must be explicitly defined to ensure that autonomous vehicles become part of a functionally interoperable and mutually supportive community, as opposed to being another example of ineffectively implemented technology that simply adds to the cacophony of independent and competing information sources.

Establishing data structures that define and specify the behavior of autonomous vehicles will allow software mechanisms to achieve interoperability. Without this pro-active approach, incompatibility and lack of interoperability will prevail. The exemplar that is introduced in this chapter addresses the human organizational limitations that arise in the employment of autonomous aircraft on the battlefield.

The hypothetical situation is one in which a commander has on hand a number of unmanned aerial vehicles (UAVs) that can provide surveillance and target acquisition. Because these vehicles have limited range and flight time, and because support systems must be deployed to launch and retrieve them, difficulties arise in the effective employment of this asset. There are so many factors that play into the anticipation, emergence and the satisfaction of aerial surveillance and targeting requirements, that it is at best a hit-and-miss proposal to achieve the goal of “eyes-on-target’ in a given situation.

C. EXEMPLAR: UNMANNED AUTONOMOUS AERIAL VEHICLE (UAAV) CONTROL

1. Background

It is clear that the potential for military exploitation of Unmanned Aerial Vehicle (UAV) technology is great. Advantages include “the ability to maximize available manpower, to remove personnel from unnecessary harm, and to increase situational awareness, lethality, survivability, and mission effectiveness.”[89] Capability gaps in current systems are identified by the Office of Naval Research as overly human operator intensive control, having a limited situational awareness, high bandwidth requirements, limited capabilities for communications loss, limited fault tolerance, limited multi-vehicle coordination, and a limited ability to operate in all types of airspace.[90]

The use of Artificial Intelligence(AI) autonomous agents is being considered to enhance UAV capabilities. The problem of employing multiple units requires the implementation of a community of autonomous agents that are capable of communicating amongst each other, and making cumulative determinations for mission assignment. UAVs that use autonomous technologies are referred to in this work as Unmanned Autonomous Aerial Vehicles (UAAV).

[89] Office of Naval Research, Broad Area Announcement 02-024, ”Development and Demonstration of Intelligent Autonomy in Unmanned Vehicles” September 2002.

[90] Ibid. 88

The Office of Naval Research Broad Area Announcement (BAA), entitled "Development and Demonstration of Intelligent Autonomy in Unmanned Vehicles," outlines in detail the capabilities that are envisioned for UAAVs. The BAA contains a "Statement of Research Need," and provides guidance on the focus for proposed research. The main functional areas are dynamic replanning/autonomous vehicle control, autonomous threat response, and distributed multi-vehicle cooperative control. Initial efforts are focused on simulation, to be accomplished using a government developed system that has a common interface to allow integration with current simulation systems. This requires modular agent objects that can presumably be extended to operational functionality. This work postulates that XML technologies are well suited for reconciling UAAV capabilities with the needs of the warfighter.

To this end, it is important to establish a common ontology[91] that can be applied by a variety of unmanned vehicles in the military environment. This ontology must take into account the principles of Artificial Intelligence (AI) and Autonomous Agent theory, as well as the constraints and requirements of military operations and personnel. This work introduces a methodology that uses the Extensible Markup Language (XML) to express this ontology and to provide a tangible set of data structures and basic computational processes that can be implemented by different UAAV control systems.

2. Focus

In order to design a system that will support autonomy in UAV technology it is necessary to distinguish a pattern of behavior that can be ascribed to agents using an established ontology. This requires the description of "*ontological commitments* for a set of agents so that they can communicate about a domain of discourse without necessarily operating on a globally shared theory."[92]

[91] "A specification of a conceptualization" T. R. Gruber. "A translation approach to portable ontologies." *Knowledge Acquisition*, 5(2): 199-220, 1993.

[92] Gruber , T. R. "Toward principles for the design of ontologies used for knowledge sharing." March 1993.

Because the military communication environment is often austere, and because the complex human interface devices can be difficult to implement in the military environment, it is important to be explicit, uniform, and extensible in these descriptions.

There will be a significant variance in behavioral theories for different UAAVs in the spectrum of possible missions. These theories will change and adapt as new technologies establish new capabilities. A well-designed ontology will allow the expansion of UAAV technology instead of repetitive reinvention. The UAAV language, like all languages, will also expand, but the standards and procedures that govern XML will allow programmatic adjustment so that systems will be able to communicate using any iteration of the ontology. One of the mechanisms that allows this adaptability is XSLT. A UAAV ontology must accommodate diversity and change due to the fluid and rapidly advancing nature of AI based autonomous technologies.

An overriding goal of autonomy in UAAVs is to elicit complex behavior using an iterative computational model that applies a simple control set to a simple set of definitions that describe a UAAV. The standardization of the control set, and the UAAV description set is a fundamental requirement for unified forward progress.

An important requirement in the military environment is understandability. Leaders are required to make decisions that can cause loss of life, and possible success or failure in battle. The role of UAAV technology in battlefield decision-making will be significant, and it is important that the control mechanisms and response characteristics of a UAAV system be easily understood and mastered. The power of computer technology to encapsulate and transform information must be applied in such a way as to allow efficient control, without permitting false impressions or erroneous results. The difficulty of this enterprise is such that it will undoubtedly need a great deal of adjustment as time progresses.

3. Qualitative Physics

The AI theory of Qualitative Physics (QP) can be used to develop a set of qualitative states and state transitions to describe the activities of UAAVs. An analogy to this approach is exemplified by the controls of an automobile. A driver does not have to know all the workings of the engine, transmission and braking system in order to control the vehicle, because all of the necessary information and control devices are standardized in the steering wheel, pedals, levers, and dashboard indicators. Standardized control and feedback mechanisms such as this will provide a focus for the development and improvement of UAAV systems.

Forbus describes a QP theory in which all possible behaviors of a physical system can be graphed in an "envisionment diagram," and that criteria can be defined to translate numerical data into a qualitative description of a characteristic or factor.[93] This technique can be applied to aeronautical information such as lift, drag, payload, fuel consumption and fuel capacity so that the range of a UAAV can be expressed as a unit of time. The quantity space of such a QP device consists of the aforementioned constants, as well as the variables introduced by terrain mission requirements and emergent environmental conditions. The result is a qualitative calculation of flight time remaining. A military decision process is best supported by the basic knowledge of how long a particular UAAV can stay aloft in a particular situation, without having to consider the underlying details. This is an example of a fact that can be asserted in the ontology, and that UAAVs can be required to report in a uniform manner.

An important aspect of QP, Qualitative Simulation (QSIM), can be used to both predict potential outcomes of decisions, and to graphically display them for analysis before implementation. All qualitative simulation systems predict multiple possible behaviors given certain sets of qualitative constraints and initial conditions.[94] A qualitative model is described by Kuipers as a "set of variables

---

[93] Forbus, Kenneth D., *Qualitative Process Theory*, PhD thesis, Massachusetts Institute of Technology, 1984

[94] Kuipers, Benjamin, Qualitative simulation. In R. A. Meyers (Ed.), *Encyclopedia of Physical Science and Technology*, Third Edition, NY: Academic Press. 2001.

related by constraints." This construct can be intuitively represented using XML which computers can readily manipulate . A quantity space in this case might be represented by a set of terrain elevations, with the landmarks being maximum elevations along a given path. The similarity with Kuipers' theory on the application of qualitative differential equations is anecdotal, but the basic concept is applicable and can certainly be extended using his algorithm.

With the standardized XML defined terrain format from Chapter IV; a UAAV can upload a local terrain set from a base station and use the data to calculate mission costs and best paths before takeoff. Figure 36 shows the simplest method of terrain avoidance that can be determined before a mission is undertaken, and can be used as an upper bound for a direct route calculation. It may take less fuel to take an indirect path that does not require the maximum altitude. This is a calculation that can be determined continuously by onboard software using standard XML defined terrain and route data. Prior knowledge of terrain can be useful to many kinds of autonomous vehicles.

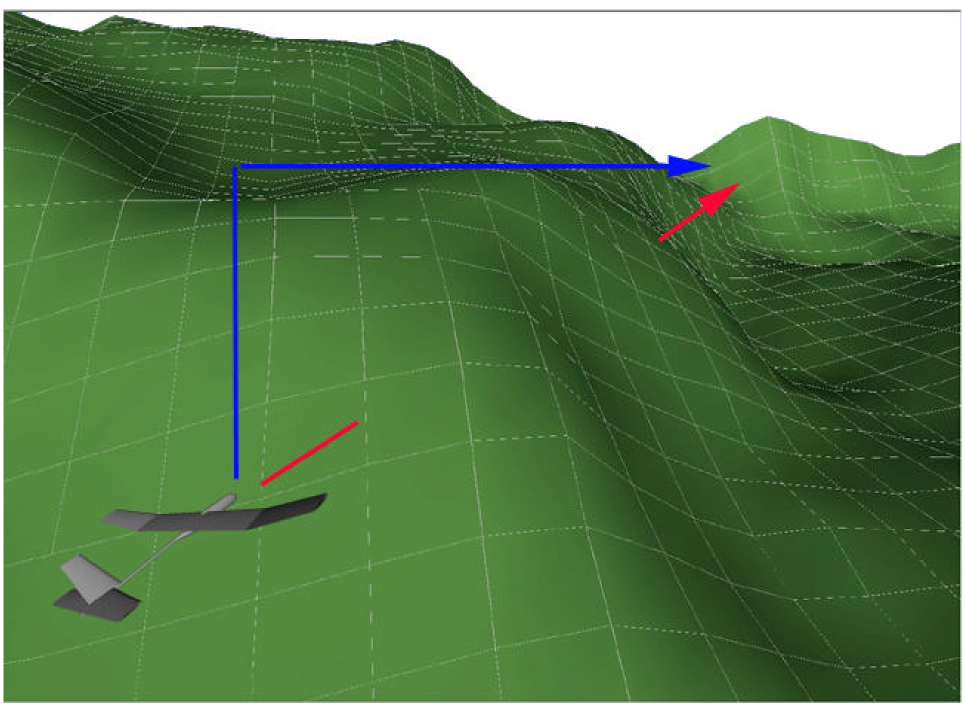


Figure 36. Terrain Avoidance: Highest Point on Path

Figure 37 illustrates an XSLT driven procedure that determines the range of set of UAAVs in a particular mission environment, and suggests a go/no-go/or partial-go decision. Range calculation in this example is simplified to consider terrain elevation and required altitudes only. The determination of shortest path or minimum-energy routes can be accomplished using qualitative simulation with the same XSLT methodology. Such an extension can remain within the scope of the suggested ontology, while improving analysis results. In this manner, significant improvement can be achieved without reinvention. An improved XSLT script is all that is necessary. This kind of direct access to algorithmic procedures without the difficulties of re-factoring code allows experimentation, extension, and improvement.

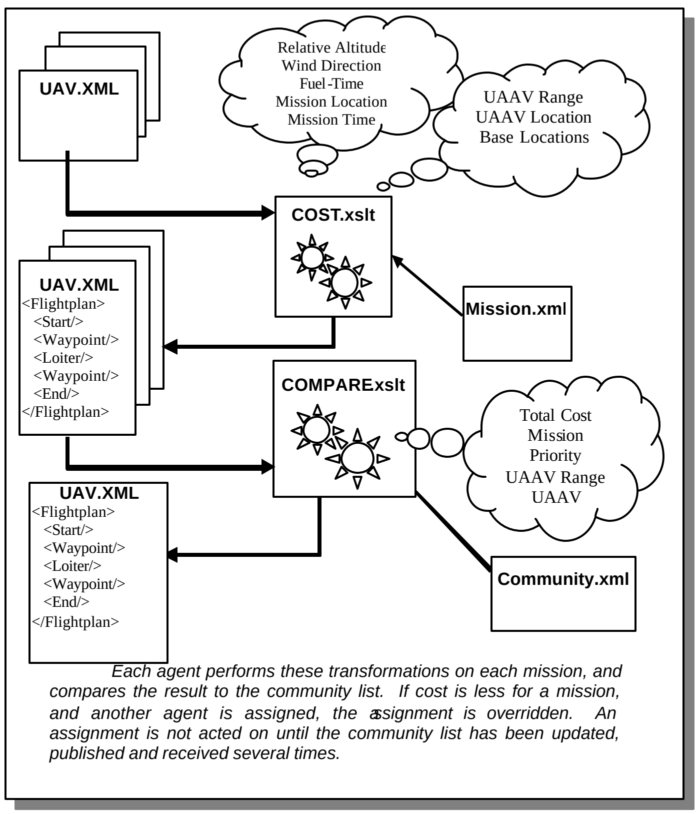


Figure 37. Mission Assignment by Data Transformation

The Qualitative Physics approach in the development of practical agent based systems is only part of the solution. The environment of a UAAV is highly variant and unpredictable. Feedback and emergent behavior must be exposed and manipulated in such a way as to accommodate the effects of weather, wind, uncharted obstacles, communications anomalies, and human error. A UAAV that faces a consistent headwind will experience fuel depletion that might prevent mission accomplishment. The unit must be able to modify or abort its flight plan independently. If possible, it may need to allow another UAAV to fulfill the request for support. Obstacles such as buildings, trees, and wind conditions must become a part of a regularly updated collective knowledge base so that they can be considered in future calculations. A fully adaptive system requires the principles of multi-agent system design.

4. Multi-Agent Systems

Characteristics of Multi-Agent Systems (MAS) are that (1) each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint; (2) there is no system global control; (3) data are decentralized; and (4) computation is asynchronous.[95] Agent Oriented Programming can be considered as a subset of Object oriented programming. The distinctions are important, and include factors such as the autonomous nature of agents, the concept of flexibility or emergent behavior, and that agents are each considered as having their own thread of control.[96]

The UAAV environment is well suited to an agent based programming methodology because it is populated by physically isolated objects. A more robust methodology might ascribe a distributed computing model to the system. The key limitations that prevent this approach are unreliable communication systems, and complex environmental factors. Autonomy and emergent behavior are not new in UAV technology, and many existing aircraft employ these principles. The use of multi-agent techniques is proposed as a way to optimize human command and control capabilities for the employment of many and varied UAAVs.

[95] Sycara, Katia, *MultiAgent Systems*, AI Magazine 19(2), 1998.

[96] Flores-Mendez, Roberto A., *Towards a Standardization of Multi-Agent System Frameworks*, ACM Crossroads Magazine, 1999

A methodology for developing a system of autonomous agents involves the definition of the factors of environment, objects, agents, relationships, operations, and laws[97]. To develop the prototype ontology, a corresponding human-machine readable XML Schema is proposed for each factor. These XML Schemas are simplified and incomplete. They represent a starting point for focus and extension. XML Schemas represent the primary tool for experts and designers to express necessary components of an applied technology.

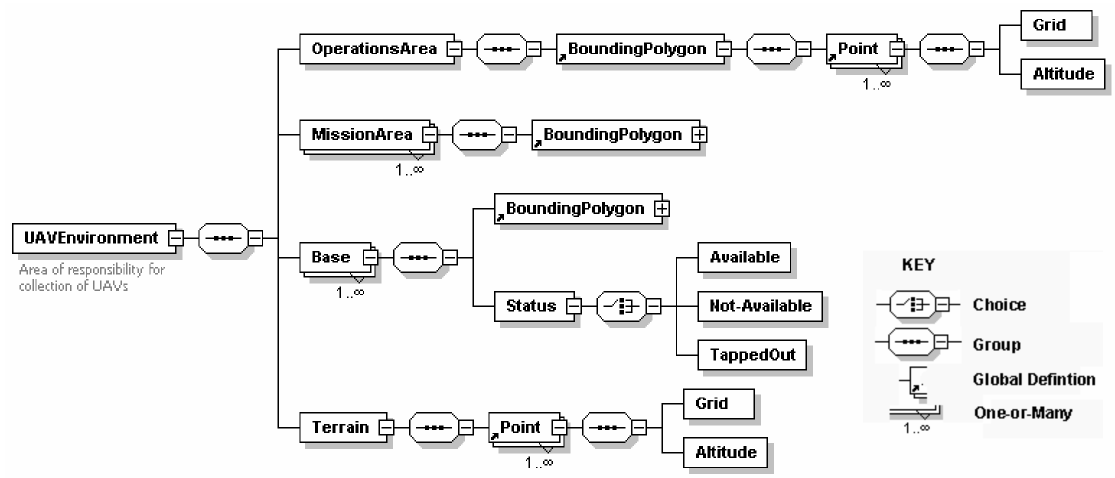


Figure 38. Prototype UAAV Environment Schema

a) Environment

Figure 38 is a diagram of the UAAV Environment Schema. The environment of a UAAV Agent is an operational area, its range, identified areas of interest, sea, land or air based refueling stations, and sea, land or air based maintenance bays. It can be expressed mathematically as spatial terrain features, and objectively as a set of targets and bases.

[97] Ferber, Jacques, *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999

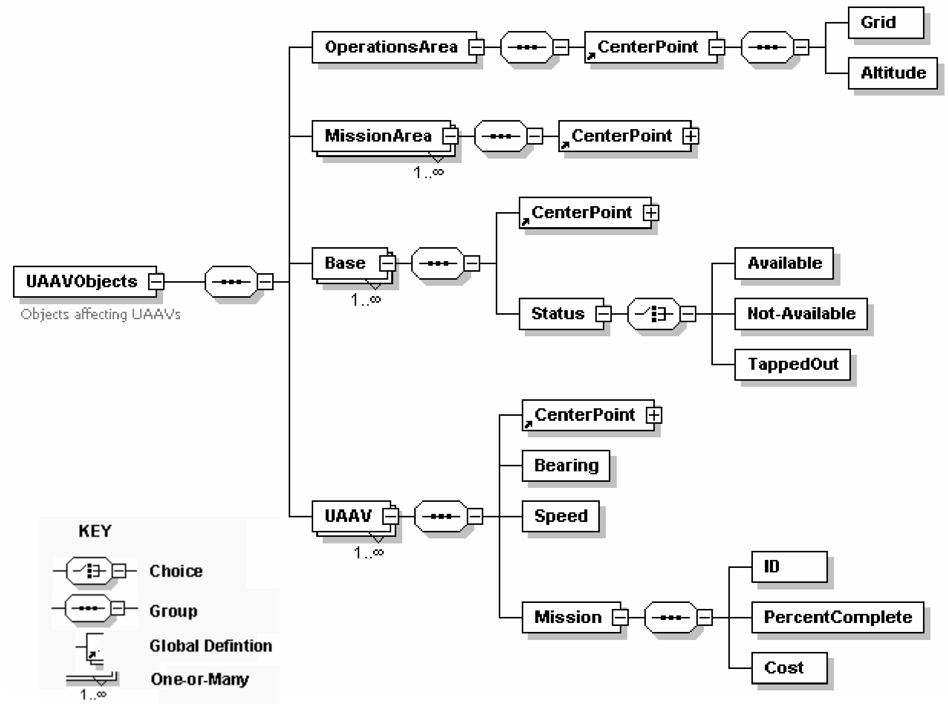


Figure 39. Prototype UAAV Objects Schema

b) Objects

Figure 39 is a diagram of the UAAV Objects Schema. The objects that a UAAV Agent must recognize include the Mission Areas and Bases as described above, as well as friendly units and equipment, enemy units and equipment, terrain features, other aircraft, weather factors, and the community of entities that can task or request support from the UAAV. The real time environment and the constant introduction of new factors contribute to the limited viewpoint that Sycara describes. There is a difference between what a UAAV knows about itself and what is published – or what it sees of other UAAVs. While there may be cases where total exposure of all data content is possible and desirable, the transmission costs and processing requirements are most likely prohibitive. Deciding what information a UAAV will process and publish is a key design factor both for hardware and for software. Principles of Qualitative Physics lead to the consideration of environment entities defined by bounding polygons, as objects defined by a center point. As in any multi-faceted environment, perspective is an important consideration.

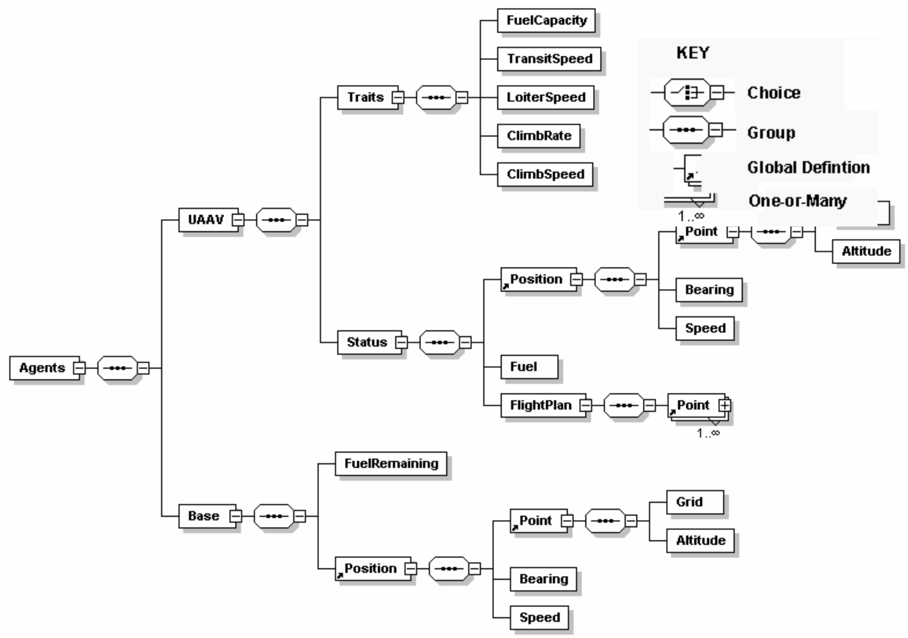


Figure 40. Prototype UAAV Agents Schema

c) Agents

Figure 40 is a diagram of the UAAV Agents Schema. A UAAV must recognize and interact with other UAAVs and base stations. The information that defines a UAAV can be extremely detailed and might concern a multitude of subjects from avionics to sensor capabilities to weapons control. This representation is rudimentary and is perhaps a minimum model for navigation, coordination and mission selection.

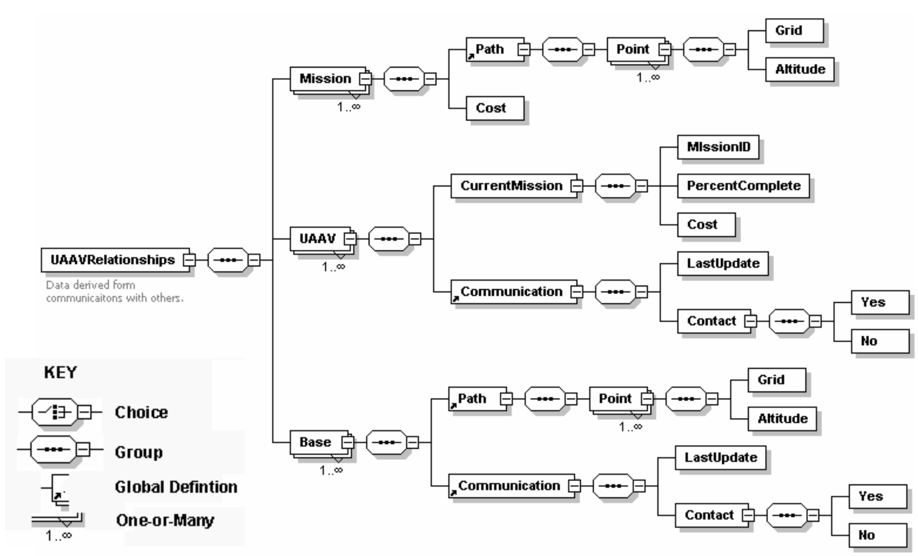


Figure 41. Prototype UAAV Relationships Schema

d) Relationships

Figure 41 is a diagram of the UAAV Relationships Schema. The relationships that a UAAV must maintain are those between customers (the community of military commanders) and between each other. Requests for service are received by any agent and added to a request list. The community of UAAVs constantly monitors this list and the most capable agents assign themselves to each mission. Capability is measured in terms of proximity, range, and fuel capacity in relation to the mission request. Response speed and density can be manipulated using mission priority, mission categorization, and area of required coverage values. A certain amount of arbitration is required among the UAAVs in the assignment of missions and is at the heart of the design problem.

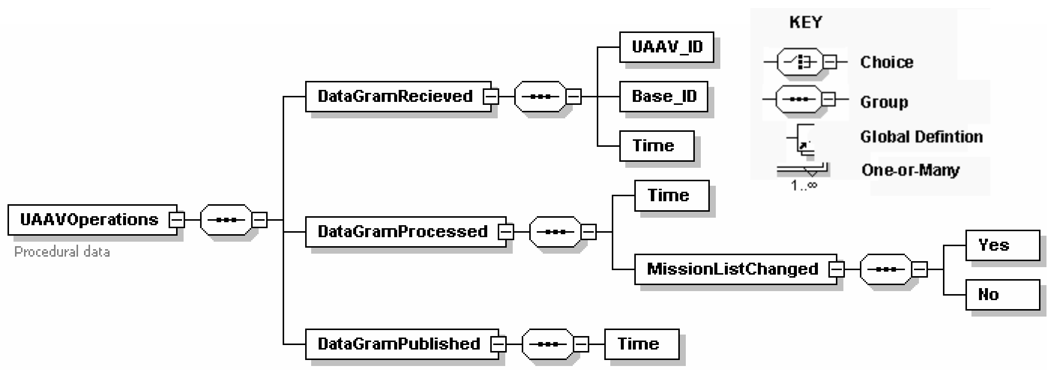


Figure 42. Prototype UAAV Operations

e) Operations

Figure 42 is a diagram of the UAAV Operations Schema. The communications limitations of this system require a very modular and simplified approach to interaction. Each agent maintains a set of data structures that describe status data, mission data, community data, and environment data. These data structures are transmitted and retransmitted as broadcast messages in a specific time orchestrated order. At any given time, each agent is ready to transmit, receive, or receive and re-transmit datagrams depending on dynamically assigned roles. These datagrams are manipulated using lightweight XSLT transformations and code in order to extrapolate missions and roles independently. As in most systems, this one will benefit from prior planning, since mission assignments can be calculated, arbitrated and assigned well in advance of takeoff. Emergent conditions will also be communicated in the datagrams so that community adjustment can occur.

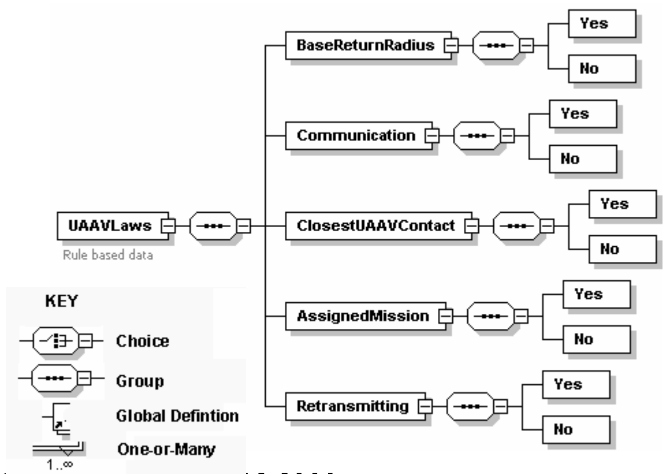


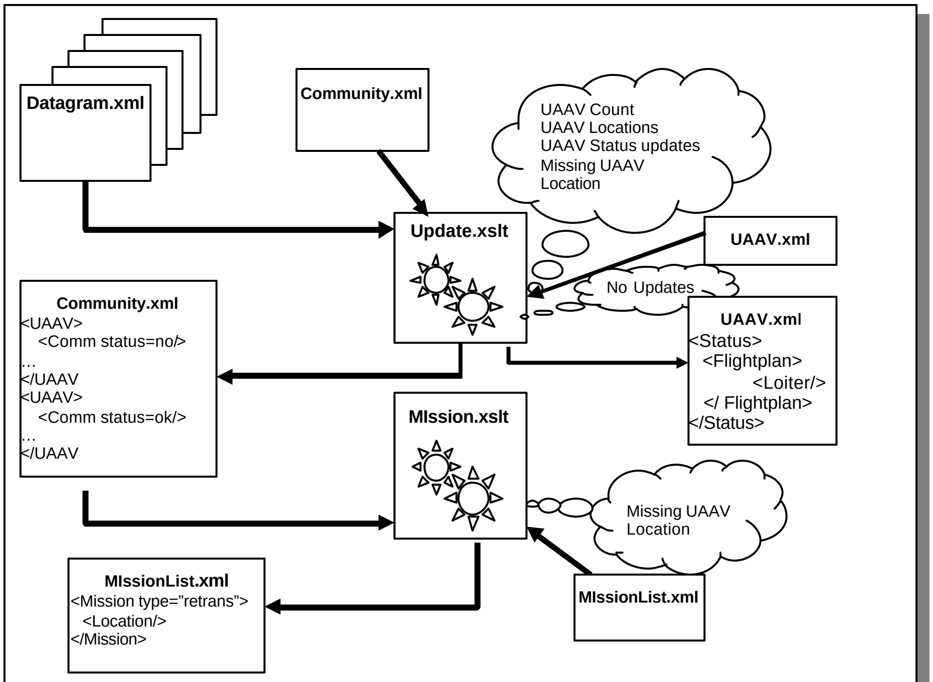
Figure 43. Prototype UAAV Laws

f) Laws

Figure 43 is a diagram of the UAAV Laws Schema. UAAV Agents operate according to a hierarchy of constraints that prioritize self-preservation and mission accomplishment in that order. At all times every UAAV maintains a current route plan that leads through self-assigned mission areas and back to a base station. UAAVs seek to maximize fuel use per flight. UAAVs will loiter when communication is lost and return to a base when fuel requirements dictate. If a UAAV loses communication, the last closest UAAV will alter flight path to maintain line of sight communication. At all times, a UAAV will maintain a calculated flight plan to a base that is within fuel consumption bounds. If these bounds are reached, return to base will be automatically triggered and current missions will be discontinued. Laws are considered as rule based elements.

g) Communications

Figure 44 shows the procedure by which the communication of data is tracked, transformed, and acted upon. Loss of communications is addressed by other agents with the addition of a retransmission mission to the mission list. An agent that loses communication with a base or other agents will assume a loiter pattern until fuel limits require a return to the last known base position.



Each Agent manufactures a Community List and MissionList from received datagrams. The datagram that is published includes requested retrans missions. These missions will often be accomplished by route alteration of airborne agents, or activation of retransmission functionality on agents that are nearest to the "missing" agent. An agent is only missing if all agents do not receive an update. If an agent doesn't receive an update, but the datagram from other agents contain updates, then it is not considered missing by the individual agent. Once a mission is determined to be necessary, the mission list is processed as in Figure 1.

Figure 44. Communications Arbitration

h) Behavior

The requirements of Multi-Agent methodology, and the techniques of Quantitative Physics can be synthesized to produce a simplified model of some basic processes that all UAAVs must be capable of. The specific definition of these processes is beyond the scope of this paper, but a basic exemplar is used to demonstrate the functionality that an XML based ontology brings. The principal benefit of XML Schema is in the form-follows-function aspect where the parameterization of procedures is used to directly perform them.

Figure 45 illustrates the anatomy of a UAAV mission, and distinguishes between the data that will be provided to the UAAV and the implied tasks that it must derive from the accumulation of factors. The cumulative effects of the data described by the governing schemas will be dynamically calculated and updated in order to reach the derived requirements.

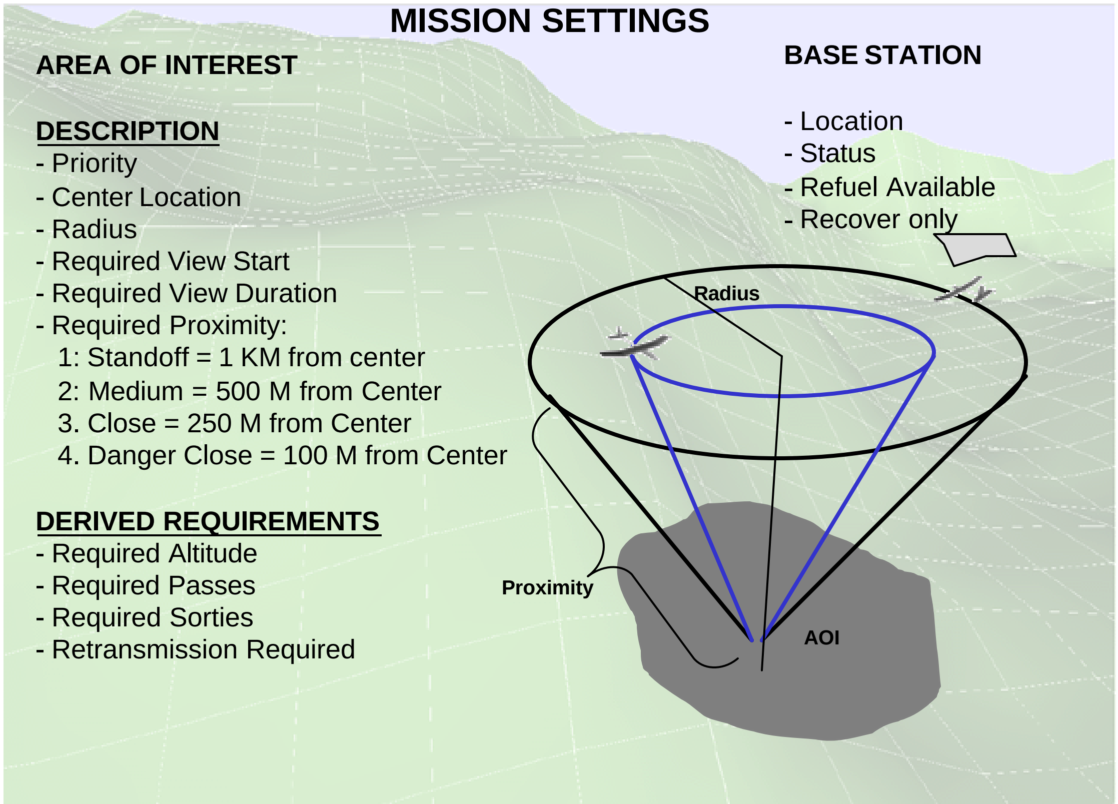


Figure 45. UAAAV Mission Parameters

This prototype model relies on a set of basic computations that each UAAV performs on a specific interval. These calculations process and create datagrams, evaluate courses of action, and monitor environmental conditions. The choice of mission is computed by comparing potential paths over known terrain given current relative location and the relative location of other UAAVs. Emergent behavior results because the last two factors change continuously.

This methodology requires terrain evaluation and the maintenance of known paths to mission areas and to bases. Prior calculations of costs along those paths are key in the evaluation of mission and for survivability. Continuous updates of these calculations must be performed to accommodate and adjust for environmental factors. For example, a UAV that is closer to a mission area, but calculates a high cost due to a headwind, will leave the mission to one that has a lower cost. Similarly, wind might play a role in the determination of return capabilities. Paths are maintained as lists of three-value strings representing grid coordinates and altitude. The process relies on quick extrapolation of known elevations between two points on a uniform grid. The XSLT code MissionData.xsl instantiates TerrainCalc.java, which implements an algorithm by Andrew Shapira, in the code GridIntersect.java.[98] A graphical representation of mission cost determination is given in Figure 46.

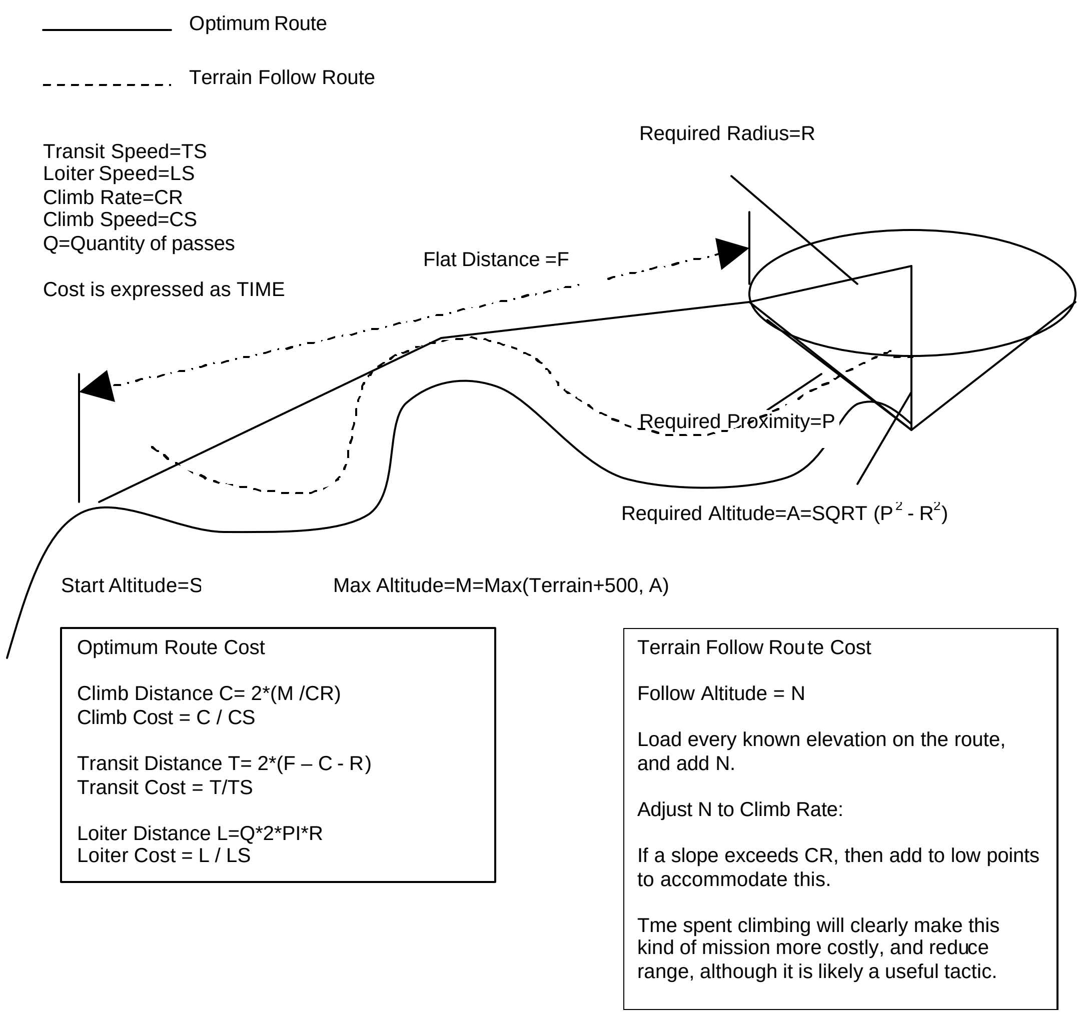


Figure 46. Mission Cost Determination

The processing cycle of an individual UAV is represented in figure 47. The example given uses a combination of XSLT and Java to extract

[98] Shapira, Andrew, *Fast Line-Edge Intersections on a Uniform Grid*, ``Graphics Gems,'' Andrew Glassner Ed., Academic Press Inc., 1990.

elevations, and calculate resulting trip cost along a line between the subject UAAV and each mission location. The initial mission request is processed and added to the MissionList.xml file using the MissionData.xsl transformation. CompareCost.xsl, compares the list of published costs form other UAAVs to decide which one is best suited for the mission. Missions are narrowed by another XSLT procedure, Eligible.xsl, that eliminates missions based on current fuel capacity. The final decision is based on mission priorities through Prioritize.xsl. This succession of transformations results in a document or XML fragment that contains a current mission selection for the UAAV. This is added to the datagram containing location and fuel state, and is published back to the community.

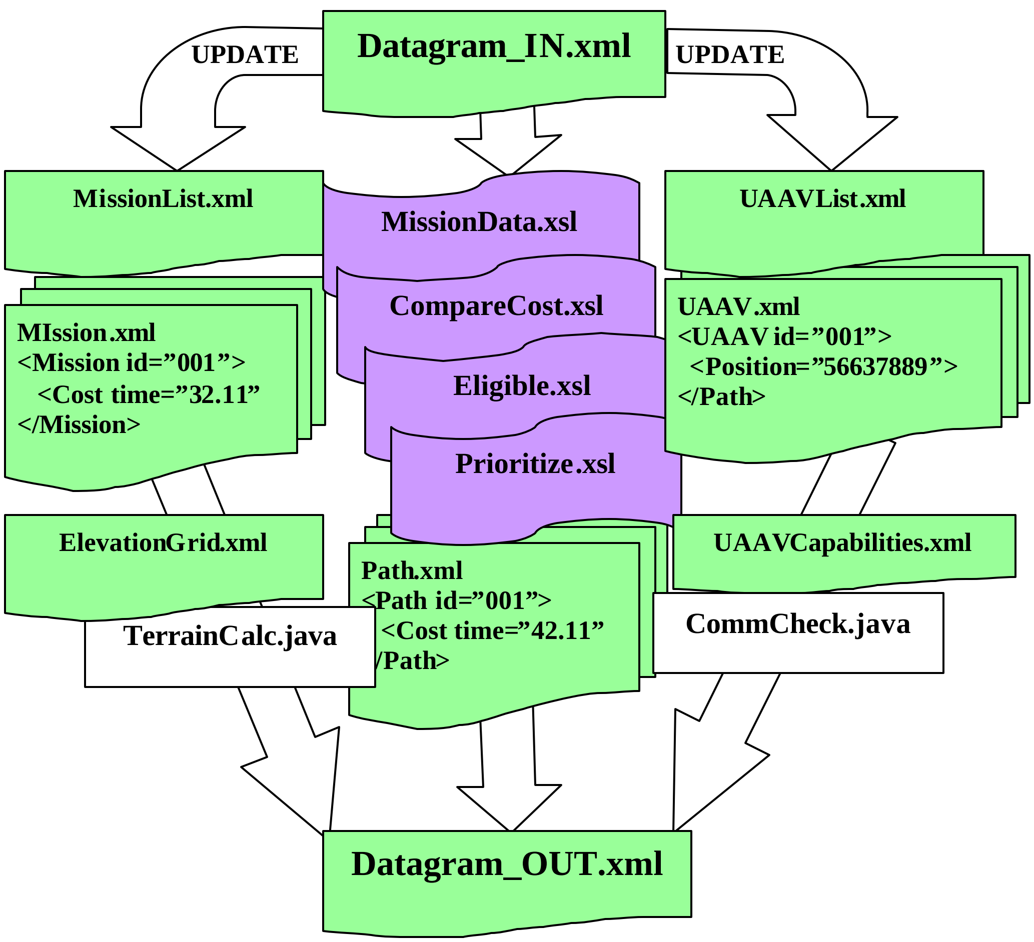


Figure 47. Agent Thought Process

5. UAAV Requirements

The requirement for dynamic replanning and autonomous vehicle control[99] is addressed by the continuous re-calculation of flight plans. The ability to perform these calculations efficiently is reliant upon the simple expression of flight characteristics and the representation of terrain as a data set that can be used to calculate landmarks. The requirement for autonomous threat response[100] is only addressed in this context for the case of lost communications. Extension of the XML documents that define UAAV capabilities and characteristics will allow the integration of more detailed threat response algorithms without affecting the current system.

Distributed multi-vehicle cooperative control[101] is accomplished by the Multi-Agent System design approach and by the implementation of XSLT transformations. Algorithms that improve the XSLT processes dynamically can be developed and updated XSLT transformations can also be distributed through the broadcast datagram system. This will allow emergent learning behavior, and the application of advanced algorithms to enable such things as real-time terrain mapping, local area network support, close air support, and participation as active weapons systems in full scale combined arms operations.

D. CONCLUSIONS

This chapter describes the theory and methodology required for the development of the XML Schemas, resultant XML documents, and XSLT transformations that comprise an ontology for the control of UAAV systems. The advantages of this approach are in its simplicity, human readability and expansive capacity for change. The use of a standardized open source markup language to define universal characteristics of this system will influence not only

[99] Ibid. 88

[100] Ibid. 88

[101] Ibid. 88

the development of software tools, but will also provide hardware developers with important guidelines for software integration development.

As basic characteristics and requirements are defined and expressed, design parameters for UAV technologies will follow a conformist trend that will allow different systems to operate in a cooperative environment. The ideal environment is one in which the limitations of one system are compensated by the advantages of another.

The diversity that arises from competition and technological innovation is not always constructive. It remains the responsibility of military professionals and military academic communities to create standards that encourage structured diversity, which combines to perform as a force multiplier across the spectrum of modern warfare. This exemplar is meant as a starting point for the exercise of leadership Data Control over autonomous entities on the battlefield.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSIONS AND FUTURE WORK

A. RECOMMENDATIONS FOR FUTURE WORK

1. Data Control

This thesis advocates the rejection of proprietary software formats, primarily in the area of Office applications. Future work in this area will be the adoption and extension of the OpenOffice.org native XML formats and base application. This software is ready for branding and customization for service and community specific tasks. OpenOffice can be customized to accommodate all military clerical, planning and operational tasks by building specialized forms and templates into the application.

All letter formats, report formats, rosters, and other documents will be uniform and functional across the enterprise, and will be designed to collect all information as XML Schema governed structured data for distribution on the GIG. Resources that are currently spent on licenses will be spent on development and support for this software. This hypothetical "MilOffice" will come in service and organization specific flavors that will be available to all on the GIG. Customization will be possible using XML and XSLT. All XML Schemas will be defined and ratified by leadership.

2. XML Tools

The toolset that was developed to accomplish the goals of this thesis is Java specific. The classes are generally simple and generic. Improvement of this code, and comprehensive documentation using Java Doc will make this toolset more useful. Developers that implement, extend, or improve the software from this thesis are encouraged to build upon the XML Tools methodology for isolating the XML specific code as a way to accommodate change when it occurs.

3. Terrain

The most developed and immediately useful exemplar in this work is the DTED Terrain Server. The potential improvements that can be applied to this project are limitless. Better navigation techniques, and viewpoint control using voice commands are interface improvements that are being considered. The implementation of the same methodology to expose raster based data such as maps, and photo imagery is impending. The establish of a Bathymetry database for naval use is also in the works.

The extension of web delivered 3D terrain views to command and control software and planning tools has great potential. The representation of communications capabilities in 3D has been addressed in a previous NPS thesis, and can be applied to assist in planning for military communication networks. Integration of battlespace symbology and interface tools will allow the use of 3D terrain as a “digital sand table.”

Concurrent thesis work conducted by Major Steve Grass, USMC, accomplished the creation of Fire Plan Sketches in the USMC command and control software C2PC. Because 3D terrain has significant tactical impact, it is the most appropriate context in which to render fire plan sketches at the platoon and company level. Efforts to accomplish this are underway.

An NPS Thesis by Major Claude Hutton, USMC addressed the battlespace visualization of intelligence databases. This work can also be integrated with the terrain server, in much the same way that the “battlespace aware” terrain views were developed. Just as a scene can populate itself in response to messages on the network, so can it access a database and provide a visual interface for the information.

Integration with the NIMA developed Joint Mapping Tool Kit (JMTK) is also a possibility, given that it is the basis for the Commercial Joint Mapping Tool Kit (C/JMTK). This solution delivers terrain as a native web service, as opposed to the proprietary web extension that the C/JMTK offers. Extending and improving on the web based 3D system in this project will likely provide Network-Centric solutions with better functionality than the C/JMTK.

4. Position Reporting Language

Position reporting capabilities are jealously guarded by telecom industries. It is a logical fact that position reporting can be accomplished using any wireless gear, with or without GPS, as long as certain information is communicated over a network. Important work in this area is in the development of military acquisitions and development requirements that define and mandate specific data transfer principles. The PRL as designed is suitable for embedding in hardware and can be developed as a standalone product for use by any piece of gear.

5. Battlespace Information Exchange

Although not as spectacular as the terrain server, the development of BIEML as an expression of the C2IEDM is an accomplishment that can be leveraged on many different levels. The most prominent effort in this area is the development of the Battle Management Language (BML).[102] This project follows principles that are compatible with those set forth in this thesis, and promote the convergence of command and control ontologies with modeling and simulation ontologies. Using XML Schema to define ontologies, and to express languages that are currently defined in field manuals is an important theme of the BML work.

The use of network chat has become a new communications tool in the battlespace, and efforts are ongoing to model tactical chat messages using the Battlespace Generic Hub so that information can be structured and controlled. The integration of tactical chat and message formats with the C2IEDM requires the development of XML Schemas that express all of the existing standard message formats. XSLT processes must then be designed, and tested so that they consistently publish appropriate data to the articulation ontology which is the C2IEDM.

6. Autonomous Agents

The treatment of this problem in the thesis is comparatively incomplete. Future work requires the completion of the code and XSLT processes to render operational models of the UAAVs on the 3D terrain. The code and XML in the code package provides an 80% solution to this problem, which might be an entire thesis in itself. The extension of the terrain server to render UAAVs in response to messages is already available using the JCATServer mechanism, so the only

[102] Carey, Scott A., Kleiner, Martin S., Hieb, Michael R. and Brown, Richard, *Standardizing Battle Management Language; Facilitating Coalition Interoperability*, Proceeding of the 2d Battle Management Language Symposium, 21-22 Aug 02

remaining problem is to finish the simulation of the agent thought processes and communications.

B. CONCLUSIONS

Network-Centric goals of interoperability and information exchange in the battlespace are valid, and achievable. The techniques described in this thesis exemplify extensible solutions to real problems. Adaptation of these principles is encouraged for the effective implementation of Network-Centric warfare strategies.

This thesis addresses the subject of battlespace visualization from a command leadership perspective. Data Control is introduced as a concept that recognizes the implicit responsibility of leadership to define the data formats that are the basis for all information technology operations. Limitations of current software architectures are identified and solutions for Network-Centric architectures are proposed. The need for a "mission-type, execution focused approach" as voiced by Michael Wynne,[103] is addressed by the assertion that Data Control must be manifested in the exhaustive development of XML Schemas that model specifications, doctrine, directives, and procedures. Interoperability is addressed by the development of XSLT transformations to reconcile these XML Schemas with universal schemas that comprise common articulation ontologies.

Michael Wynne cited battlespace management shortfalls in OIF that "(had we) been fighting a more competent, determined enemy ..could have been disastrous … cost lives .. delayed victory ..offered our enemies military and political opportunities to seize the initiative .. complicated the political and coalition dimensions of our operations …helped to erode public support for the operation, (and) contributed on several different occasions to the tragedy of fratricide."[104] The interoperability that is a pre-requisite for battlespace visualization is non-existent. This thesis advocates the rejection of the

103 Ibid. 22

104 Ibid. 22

proprietary software that has failed to provide interoperability and the adoption of open standards and open source software that is service vice product based.

The use of XML to force interoperability with Data Control is discussed and demonstrated in this thesis, but will not become a reality for the DoD until change is instituted, rather than just discussed. The myopic addiction to Microsoft products in the DoD is a serious problem since it is the primary toolset that is currently used for battlespace visualization. Microsoft has already instituted proprietary limitations on the use of XML[105], and will continue to maintain de-facto Data Control in order to maintain customer lock-in. For those who like to tout "Transformation" prerogatives, the word of the day is "put up or shut up," because Microsoft will not allow Data Control.

The work in this thesis examines not just the principles and theories behind Network-Centric Warfare, but the actual steps that must be taken to accomplish it. The exemplars address a cross section of the battlespace visualization problem space, and provide working examples and starting points for further development.

There are many references to leadership responsibilities in this thesis. The purpose for this is to reinforce the military context of this work, and to emphasize the requirement for leadership commitment to Data Control. Leaders at all levels must demand interoperability, reject proprietary limitations, and communicate the warfighter perspective by establishing ontologies that reflect the needs of their organization, culture, and mission.

[105] Wilcox , Joe, *Microsoft limits XML in Office 2003*, CNET News.com, ZDNN, http://www.zdnn.com., April 11, 2003

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. SOURCE CODE ACCESS AND DESCRIPTION

A. ACCESS

This thesis, references, and all source code can be obtained from:

http://terra.cs.nps.navy.mil/SavageProjects/Students/Neushul/Work.html

Additionally, access to source code and distribution may be obtained from:

James D. Neushul: jdn_email@fastmail.fm

Dr. Don Brutzman: brutzman@nps.navy.mil

Research Associate Curt Blais; clblais@nps.navy.mil

B. SOURCE CODE DESCRIPTION

The source code that is intended to accompany this thesis is in a directory named Source Code. It is divided into two subdirectories; Exemplars, and Extras. The Exemplars folder contains the code that is directly referenced in the thesis. The Extras folder contains similar work that is not discussed, but which supports the same concepts and approaches.

1. Exemplars Directory

a) Battlespace Information Exchange Markup Language

- C2IEDM Specification:

Contains the ATCCIS C2IEDM documentation from which the Battlespace Information Exchange Markup Language is extracted. Sample data for implementing the ARM database is included as well.

- Documentation:

Contains an HTML documentation package that was auto generated from the BIEML Schema. This allows in-depth exploration of the schema and data structure using a web browser.

- Schema:

Contains the schemas that are included in Appendix B, the full BIEML Schema, (BattlespaceInformationExchangeSchema.xsd), as well as the original Schema developed by Francisco Laoiza, (GH5Complete.xsd).

- ***XML:***

Contains the XML documents extracted form the individual appendices of the ATCCIS C2IEDM documentation.

- ***XSLT:***

Contains the XSLT scripts included in Appendix B. Also includes an XML file, ContentPath.xml, which is a convenience mechanism that is accessed by the XSLT to discover the locations of the documentation and XML files.

b) Position Reporting Language

This directory contains a small web application that is designed to process PRL messages sent to a Java Servlet, and make entries into a database using a SQL command. The Servlet is in the WEB-INF/classes directory. This is an abbreviated instantiation of Network-Centric position reporting procedures.

c) Terrain Servers

- ***XGLOBEServer***

Contains the XML and Java Servlet based application that is currently deployed to the internet . 3D terrain data can be obtained over the web from a password protected site: http://terra.cs.nps.navy.mil/XGLOBEServer/XGlobeSite/XGlobe.html, (Accessed 2003). This methodology can be applied to the delivery of any geographically organized data.

- ***JCATServer:***

Contains a Java Servlet application that is very similar to the XGLOBEServer that is currently deployed. The JCATServer responds to bridging code that processes datagrams broadcast in a JCATS simulation exercise. The difference between this server and the XGLOBEServer is in the EntityMover.java, and VRMLConnect.java classes that make the scenes "battlespace aware." This code is included in the GEODATA/Views/localhost and GEODATA/Views/ VMASC23 folders, which reflect the two workstations that used this server during the simulation.

d) UAAV

- References:

Contains the references that guide this approach.

- Source:

Contains the Java classes that perform some of the math. This code is incomplete, and only addresses some of the more complex problems associated with path determination. Basic trigonometry needs to be implemented in the code.

- XML:

Contains the XML instances of the Agents, the Schemas that define the Multi-Agent system, and the XSLT that processes the XML to determine behavior. This XSLT was developed with a goal to avoid Java extension, and uses XSLT based math code included in the fxsl-Xalan directory. This proved to be prohibitively slow. To move forward on this project, Java extensions must be added to perform the trigonometry.

e) XMLToolset:

- XMLTools

Contains the Java classes described in Chapter III.

- XSLTransformTool

Contains a standalone implementation of an XSLT transformation utility. Useful for testing, and integration into programs that apply XSLT.

- XSLTServlet

Contains the mechanism that is at the heart of the XGLOBEServer application. Receives a Java Servlet call in the form of a URL, which communicates an XSLT file to be applied, XML files to operate on, output method, and parameters. This is used to allow interface to unlimited software power from a web browser.

2. Extras Directory

a) DistributedInteractiveSimulation

This is a well developed project that applied a similar methodology to that in Chapter V to create a Schema for the Distributed Interactive Simulation Protocol (DIS). Although this protocol has been superceded by the High Level Architecture (HLA) for M&S, the DIS protocol contains valuable semantic information that can be combined with the C2IEDM to establish meaning and determine appropriate implementations.

Instead of extracting an XML Schema from documentation, this process extracts the data from a database that contains the relationship descriptions. The resulting Schema, DISData.xsd has not been optimized with global elements, and represents a starting point. The DISHandler folder contains code that is designed to access the DISData schema to produce DIS datagrams. This addresses issues related to the creation of XML based binary transfer protocols, which is now a major focus of the XSMF project.

b) TerrainHandler

This is an early iteration of the terrain server that is described in Chapter IV. It was designed to access a single DTED disk. It fails on disks that span large area of the earth, because there is no easy way to represent the contents of the disk with a 2D interface without implementing clumsy scrolling mechanisms. The discovery of the limitations of 2D interfaces led to the natural 3D solution that represent s large areas of the globe with ease.

c) X3D Specification Conversion

This project involved the conversion of an HTML based specification to an XML based presentation based on the W3C Specification Schema. The translation from HTML to XML makes data far more useful.

d) XGLOBEStandalone

This project was developed because of the problems with web browsers on a local machine. Security concerns prevent browsers from accessing a local file system. This problem was addressed by implementing a small server on the local machine. A machine running XGLOBEStandalone can extract, store and view DTED locally, and can provide the same service that XGLOBEServer provides to other machines. The concept that every computer can have server functionality is one that promotes Network-Centricity[106].

[106] Cebrowski, Network -Centric Warfare.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. DATABASE-TO-XML SCHEMA SOURCE CODE

1. BattlespaceInformationExchangeEntityDefinitions.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="BattlespaceInformationExchangeEntityDefinitions">
    <xs:annotation>
      <xs:documentation>Battlespace Information Exchange Entity
Definitions  Extracted using XSLT from OpenOffice XML version of ANNEX
B.  DATA MODEL DOCUMENTATION—ENTITY DEFINITIONS AND ATTRIBUTES from the
Army Tactical Command and Control Information System (ATCCIS) Working
Group Working Paper 5-5, Edition 5.0: The Land Command and Control
Information Exchange Data Model (LC2IEDM), ATTCIS Baseline 2.0, 18
March 2002.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Entity" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Attribute" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="name" type="xs:string"/>
                  <xs:attribute name="key" type="xs:string" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="name" type="xs:string"/>
            <xs:attribute name="definition" type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

2. BattlespaceInformationExchangeEntityRelationships.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="BattlespaceInformationExchangeEntityRelationships">
   <xs:annotation>
    <xs:documentation>Battlespace Information Exchange Entity
Relationships.  Extracted using XSLT from OpenOffice XML version of
ANNEX C.  DATA MODEL DOCUMENTATION—ENTITY RELATIONSHIPS from the Army
Tactical Command and Control Information System (ATCCIS) Working Group
Working Paper 5-5, Edition 5.0: The Land Command and Control
```

```
Information Exchange Data Model (LC2IEDM), ATTCIS Baseline 2.0, 18
March 2002.</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:sequence>
      <xs:element name="ParentEntity" maxOccurs="unbounded">
       <xs:complexType>
        <xs:sequence maxOccurs="unbounded">
          <xs:element name="ChildEntity">
            <xs:complexType>
             <xs:sequence>
              <xs:element name="LogicalForeignKey" maxOccurs="unbounded"/>
             </xs:sequence>
             <xs:attribute name="name" type="xs:string"/>
             <xs:attribute name="verbPhrase" type="xs:string"/>
             <xs:attribute name="relationship" type="xs:string"/>
             <xs:attribute name="cardinality" type="xs:string"/>
             <xs:attribute name="nulls" type="xs:string" use="optional"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string"/>
       </xs:complexType>
      </xs:element>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
</xs:schema>
```

3. *BattlespaceInformationExchangeAttributeDefinitions.xsd*

```
<?xml version="1.0" encoding="UTF-8"?>
<-- JD Neushul, Naval Postgraduate School-->
<xs:schema                    xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="BattlespaceGenericHubAttributeDefinitions">
    <xs:annotation>
      <xs:documentation>Battlespace Generic Hub Tactical Markup Language
Attribute  definitions.   Extracted  using  XSLT  from  OpenOffice  XML
version  of  ANNEX  D.   DATA  MODEL  DOCUMENTATION—ATTRIBUTE  DEFINITIONS
from the Army Tactical Command and Control Information System (ATCCIS)
Working  Group  Working  Paper  5-5,  Edition  5.0:  The  Land  Command  and
Control Information Exchange Data Model (LC2IEDM), ATTCIS Baseline 2.0,
18 March 2002.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Attribute" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="UsingEntity" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="name" type="xs:string"/>
                  <xs:attribute name="use" type="xs:string"/>
                </xs:complexType>
              </xs:element>
```

```
        </xs:sequence>
        <xs:attribute name="name" type="xs:string"/>
        <xs:attribute name="definition" type="xs:string"/>
      </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

4. BattlespaceInformationExchangeEnumeratedDomains.xsd

```
<?xml version="1.0" encoding="UTF-16"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="BattlespaceInformationExchangeEnumeratedDomains">
  <xs:annotation>
   <xs:documentation>Battlespace Information Exchange Enumerated
Domains  Extracted using XSLT from OpenOffice XML version of ANNEX E.
DATA MODEL DOCUMENTATION—SPECIFICATIONS OF ENUMERATED DOMAINS from the
Army Tactical Command and Control Information System (ATCCIS) Working
Group Working Paper 5-5, Edition 5.0: The Land Command and Control
Information Exchange Data Model (LC2IEDM), ATTCIS Baseline 2.0, 18
March 2002.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Domain" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="Value" maxOccurs="unbounded">
        <xs:complexType>
         <xs:attribute name="name"/>
         <xs:attribute name="definition"/>
         <xs:attribute name="source"/>
         <xs:attribute name="physicalValue"/>
         <xs:attribute name="identifier"/>
        </xs:complexType>
       </xs:element>
       <xs:element name="Usage" maxOccurs="unbounded">
        <xs:complexType>
         <xs:attribute name="entity"/>
         <xs:attribute name="attribute"/>
         <xs:attribute name="optionality"/>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
      <xs:attribute name="name"/>
      <xs:attribute name="definition"/>
      <xs:attribute name="source"/>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
```

```
</xs:schema>
```

5. ExtractEntityDefinitions.xsl

```
<?xml version="1.0"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xsl:stylesheet version="1.1" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:office="http://openoffice.org/2000/office"
xmlns:table="http://openoffice.org/2000/table"
xmlns:text="http://openoffice.org/2000/text"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" exclude-result-
prefixes="office table text">
  <xsl:output method="xml" indent="yes"/>
  <xsl:strip-space elements="*"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <!--  Then param 'SourceDocPath' is uses as a convention to assign the path to
the source content of the C2IEDM specification.  All of these XSLT scripts use
this convention so that the location of the source content can be changed in
one document, the ContentPath.xml document which is assumed to be in the same
directory.-->
  <xsl:param name="SourceDocPath" select="document('ContentPath.xml')/Paths"/>
  <!-- The param 'AnnexBTable' is defined using an XPath expression that
extracts the table in which the previous header text (text:h) as the content
'ANNEX B. '.  The [1] ensures that this is the first instance of such a table.
Use of a param instead of putting the XPath directly into the apply-templates
allows this Stylesheet to be called externally with an pre-determined
parameter.  This allows a chained transformation operation for full
automation.-->
  <xsl:param name="AnnexBTable"
select="document($SourceDocPath/@content)/office:document-
content/office:body/table:table[preceding-sibling::text:h/text()='ANNEX B.
'][1]"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="/">
    <!-- Because this is generating a document that is to conform to a
predefined schema, the schema is assigned.  This assumes that the schema is in
a directory called 'Schema' on the same level ("../Schema/")  If no Schema is to
be applied then the assignment can be omitted.-->
    <xsl:element name="BattlespaceInformationExchangeEntityDefinitions">
      <xsl:attribute name="xsi:noNamespaceSchemaLocation">

  <xsl:text>../Schema/BattlespaceInformationExchangeEntityDefinitions.xsd</xsl:t
ext>
      </xsl:attribute>
      <xsl:comment>Battlespace Information Exchange Entity definitions.
Extracted using XSLT from OpenOffice.org XML version of  ANNEX B.  DATA MODEL
DOCUMENTATION—ENTITY DEFINITIONS AND ATTRIBUTES from the Army Tactical Command
and Control Information System (ATCCIS) Working Group Working Paper 5-5,
Edition 5.0: The Land Command and Control Information Exchange Data Model
(LC2IEDM), ATTCIS Baseline 2.0, 18 March 2002.</xsl:comment>
      <xsl:apply-templates select="$AnnexBTable/table:table-row"/>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="table:table-row">
    <!--Create a new element named 'Entity' for each row-->
    <xsl:element name="Entity">
      <!--Extracts the first cell of the row and assigns value to attribute
name-->
      <xsl:attribute name="name">
        <xsl:value-of select="normalize-space(table:table-cell[1])"/>
      </xsl:attribute>
```

```
      <!--Extracts the second cell of the row and assigns value to attribute
definition-->
      <xsl:attribute name="definition">
        <xsl:value-of select="normalize-space(table:table-cell[2])"/>
      </xsl:attribute>
      <!--The third cell of the table contains several entries, each of which is
contained in a text:p element.  The contents are database Attributes (not to be
confused with XML attributes) .. Create a child element called 'Attribute' for
each one-->
      <xsl:for-each select="table:table-cell[3]/text:p">
        <xsl:element name="Attribute">
          <!--Inside each Attribute definition there is information that pertains
to the type of database key.  Use another template to add attributes 'name'
and' key' that clearly reflects this-->
          <xsl:call-template name="parseAttributeText">
            <xsl:with-param name="AttText" select="normalize(.)"/>
          </xsl:call-template>
        </xsl:element>
      </xsl:for-each>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template name="parseAttributeText">
    <xsl:param name="AttText"/>
    <!-- This amounts to a text parsing algorithm that relies on the content to
convert "free text" into structured data this will break if the text is entered
differently into the table.  Currently, the data is entered very consistently
so this works OK.  A table design that better depicts this information would be
preferred-->
    <!-- Add an XML attribute 'name' to the XML Element 'Attribute'-->
    <xsl:attribute name="name">
      <xsl:choose>
        <xsl:when test="contains($AttText,'(')">
          <xsl:value-of select="translate(substring-before($AttText,'('),'
',''))"/>
        </xsl:when>
        <xsl:when test="not(contains($AttText,'('))">
          <xsl:value-of select="translate($AttText,' ','')"/>
        </xsl:when>
      </xsl:choose>
    </xsl:attribute>
    <!-- Add an XML attribute 'key' to the XML Element 'Attribute'-->
    <xsl:choose>
      <xsl:when test="contains($AttText,'(PK)')">
        <xsl:attribute name="key">
          <xsl:text>primary</xsl:text>
        </xsl:attribute>
        <xsl:if test="contains($AttText,'(FK)')">
          <xsl:attribute name="key">
            <xsl:text>primary, foreign</xsl:text>
          </xsl:attribute>
        </xsl:if>
      </xsl:when>
      <xsl:when test="contains($AttText,'(FK)')">
        <xsl:attribute name="key">
          <xsl:text>foreign</xsl:text>
        </xsl:attribute>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
</xsl:stylesheet>
```

6. *ExtractEntityRelationships.xsl*

```
<?xml version="1.0"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xsl:stylesheet version="1.1" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:office="http://openoffice.org/2000/office"
xmlns:table="http://openoffice.org/2000/table"
xmlns:text="http://openoffice.org/2000/text"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" exclude-result-
prefixes="office table text">
  <xsl:output method="xml" indent="yes"/>
  <xsl:strip-space elements="*"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <!--  The param 'SourceDocPath' is uses as a convention to assign the path to
the source content of the C2IEDM specification.  All of these XSLT scripts use
this convention so that the location of the source content can be changed in
one document, the ContentPath.xml document which is assumed to be in the same
directory.-->
<xsl:param name="SourceDocPath" select="document('ContentPath.xml')/Paths"/>
  <!-- The param 'AnnexCTable' is defined using an XPath expression that
extracts the table in which the previous header text (text:h) as the content
'ANNEX C. '.  The [1] ensures that this is the first instance of such a table.
Use of a param instead of putting the XPath directly into the apply-templates
allows this Stylesheet to be called externally with an pre-determined
parameter.  This allows a chained transformation operation for full
automation.-->
  <xsl:param name="AnnexCTable"
select="document($SourceDocPath/@content)/office:document-
content/office:body/table:table[preceding-sibling::text:h/text()='ANNEX C.
'][1]"/>
  <xsl:template match="/">
    <xsl:element name="BattlespaceInformationExchangeEntityRelationships">
      <xsl:attribute name="xsi:noNamespaceSchemaLocation">
        <xsl:text>
          ../Schema/BattlespaceInformationExchangeEntityRelationships.xsd
        </xsl:text>
      </xsl:attribute>
      <xsl:comment>Battlespace Information Exchange Entity definitions.
Extracted using XSLT from OpenOffice.org XML version of ANNEX C.  DATA MODEL
DOCUMENTATION—ENTITY RELATIONSHIPS from the Army Tactical Command and Control
Information System (ATCCIS) Working Group Working Paper 5-5, Edition 5.0: The
Land Command and Control Information Exchange Data Model (LC2IEDM), ATTCIS
Baseline 2.0, 18 March 2002.</xsl:comment>
      <xsl:apply-templates select="$AnnexCTable"/>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="table:table">
    <!-- This table is very complex and requires some manipulation.. This
statement applies the template 'BuildParentEntityTree' to the first cell of
each row of which the content of the first cell does not equal the content of
the first cell of the preceding row.-->
    <xsl:for-each select="table:table-row[not(table:table-
cell[1]/text:p=preceding-sibling::table:table-row/table:table-
cell[1]/text:p)]">
  <xsl:call-template name="BuildParentEntityTree">
        <!--The first cell is the parent entity-->
        <xsl:with-param name="Parent" select="table:table-cell[1]"/>
      </xsl:call-template>
    </xsl:for-each>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
```

```
  <xsl:template name="BuildParentEntityTree">
    <xsl:param name="Parent"/>
    <!--Create the ParentEntity Element-->
    <xsl:element name="ParentEntity">
      <xsl:attribute name="name">
        <xsl:value-of select="normalize-space($Parent)"/>
      </xsl:attribute>
      <!-- Go back to the top of the section and get every row whose first cell
is the same as the parent-->
      <xsl:for-each select="$AnnexCTable/table:table-row[table:table-
cell[1]=$Parent]">
        <!--Create the ChildEntity Element-->
        <xsl:element name="ChildEntity">
          <!--cell #3 is the name-->
          <xsl:attribute name="name">
            <xsl:value-of select="normalize-space(table:table-cell[3])"/>
          </xsl:attribute>
          <!--cell #2 is the verbPhrase-->
          <xsl:attribute name="verbPhrase">
            <xsl:value-of select="normalize-space(table:table-cell[2])"/>
          </xsl:attribute>
          <!--cell #4 is the relationship-->
          <xsl:attribute name="relationship">
            <xsl:value-of select="normalize-space(table:table-cell[4])"/>
          </xsl:attribute>
          <!--cell #6 is the cardinality-->
          <xsl:attribute name="cardinality">
            <xsl:value-of select="normalize-space(table:table-cell[6])"/>
          </xsl:attribute>
        <!--if the text 'Allowed' is in cell #7 then nulls are allowed-->
          <xsl:if test="contains(table:table-cell[7],'Allowed')">
            <xsl:attribute name="nulls">
              <xsl:text>Allowed</xsl:text>
            </xsl:attribute>
          </xsl:if>
          <!--if the text 'No' is in cell #7 then nulls are not allowed-->
          <xsl:if test="contains(table:table-cell[7],'No')">
            <xsl:attribute name="nulls">
              <xsl:text>Not Allowed</xsl:text>
            </xsl:attribute>
          </xsl:if>
          <!--cell #5 may have no, or multiple text:p entries.  Elements will  be
created for each one or none using the template method-->
          <xsl:apply-templates select="table:table-cell[5]/text:p"/>
        </xsl:element>
      </xsl:for-each>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="table:table-cell[5]/text:p">
    <xsl:element name="LogicalForeignKey">
      <xsl:value-of select="."/>
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

7. ExtractAttributeDefinitions.xsl

```
<?xml version="1.0"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xsl:stylesheet version="1.1" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:office="http://openoffice.org/2000/office"
xmlns:table="http://openoffice.org/2000/table"
xmlns:text="http://openoffice.org/2000/text"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" exclude-result-
prefixes="office table text">
<xsl:output method="xml" indent="yes"/>
  <!--    It is important to remove white space so that content can be can be
reliably compared later. -->
<xsl:strip-space elements="*"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <!--  The param 'SourceDocPath' is uses as a convention to assign the path to
the source content of the C2IEDM specification.  All of these XSLT scripts use
this convention so that the location of the source content can be changed in
one document, the ContentPath.xml document which is assumed to be in the same
directory.-->
<xsl:param name="SourceDocPath" select="document('ContentPath.xml')/Paths"/>
  <!--  The param 'AnnexDTable' is defined using an XPath expression that
extracts the table in which the previous header text (text:h) has the content
'ANNEX D. '.  The [1] ensures that this is the first instance of such a table.
Use of a param instead of putting the XPath directly into the apply-templates
allows this Stylesheet to be called externally with an pre-determined
parameter.  This allows a chained transformation operation for full
automation.-->
  <xsl:param name="AnnexDTable"
select="document($SourceDocPath/@content)/office:document-
content/office:body/table:table[preceding-sibling::text:h/text()='ANNEX D.
'][1]"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="/">
    <xsl:element name="BattlespaceInformationExchangeAttributeDefinitions">
      <xsl:attribute name="xsi:noNamespaceSchemaLocation">
        <xsl:text>
          ../Schema/BattlespaceInformationExchangeAttributeDefinitions.xsd
        </xsl:text>
      </xsl:attribute>
    <xsl:comment>Battlespace Information Exchange Hub Tactical Markup Language
Entity definitions.  Extracted using XSLT from OpenOffice XML version of  ANNEX
D.  DATA MODEL DOCUMENTATION—ATTRIBUTE  DEFINITIONS from the Army Tactical
Command and Control Information System (ATCCIS) Working Group Working Paper 5-
5, Edition 5.0: The Land Command and Control Information Exchange Data Model
(LC2IEDM), ATTCIS Baseline 2.0, 18 March 2002.</xsl:comment>
    <xsl:apply-templates select="$AnnexDTable"/>
   </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="table:table">
   <xsl:apply-templates select="table:table-row"/>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="table:table-row">
   <!--Create a new element named 'Attribute' for each row-->
   <xsl:element name="Attribute">
    <!--It is important to ensure that spaces are stripped using normalize so
that contents can be reliably compared later-->
    <!--Extracts the first cell of the row and assigns value to attribute
name-->
    <xsl:attribute name="name">
      <xsl:value-of select="normalize-space(table:table-cell[1])"/>
```

```
    </xsl:attribute>
    <!--Extracts the fourth cell of the row and assigns value to attribute
definition
       This may seem incongruous, but it is specific to the layout of the
table-->
    <xsl:attribute name="definition">
      <xsl:value-of select="normalize-space(table:table-cell[4])"/>
    </xsl:attribute>
    <!--The third cell of the row contains complex information and needs to be
further broken down in another template-->
    <xsl:apply-templates select="table:table-cell[3]/text:p"/>
   </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="table:table-cell[3]/text:p">
   <!-- Remember the position of the text being worked on-->
   <xsl:variable name="pos" select="position()"/>
   <!-- Find out what is in cell #2 of this table row-->
   <xsl:variable name="usage">
    <!--This requires some text parsing in another template-->
    <xsl:apply-templates select="./ancestor::table:table-row/table:table-
cell[2]/text:p/."/>
   </xsl:variable>
   <!-- Create an element  called UsingEntity-->
   <xsl:element name="UsingEntity">
    <!--The name of the UsingEntity is in the current cell (#3)-->
    <xsl:attribute name="name">
      <xsl:value-of select="normalize-space(.)"/>
    </xsl:attribute>
    <!--The use of the UsingEntity is the text in cell #2 or this row that has
the same position as the text in this cell-->
    <xsl:attribute name="use">
      <xsl:value-of select="normalize-space(ancestor::table:table-
row/table:table-cell[2]/text:p[$pos])"/>
    </xsl:attribute>
   </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="text()">
   <!-- Create explicit entries in a child Element named 'use' to clarify
function of entity-->
   <xsl:choose>
    <xsl:when test=".='OP'">
      <xsl:element name="use">Optional</xsl:element>
    </xsl:when>
    <xsl:when test=".='MA'">
      <xsl:element name="use">Required</xsl:element>
    </xsl:when>
   </xsl:choose>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="text:line-break">
   <xsl:element name="blank"/>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
</xsl:stylesheet>
```

8. ExtractEnumeratedDomains.xsl

```
<?xml version="1.0" encoding="utf-8"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xsl:stylesheet version="1.1" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:office="http://openoffice.org/2000/office"
xmlns:table="http://openoffice.org/2000/table"
xmlns:text="http://openoffice.org/2000/text"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" exclude-result-
prefixes="office table text">
  <xsl:output method="xml" indent="yes"/>
  <xsl:strip-space elements="*"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-__-_-_-_-_-*-->
  <xsl:param name="SourceDocPath" select="document('ContentPath.xml')/Paths"/>
  <xsl:param name="AnnexE"
select="document($SourceDocPath/@content)/office:document-
content/office:body/table:table[preceding-sibling::text:h[1]/text()='ANNEX E.
']"/>
<xsl:template match="/">
    <xsl:element name="BattlespaceInformationExchangeEnumeratedDomains">
      <xsl:attribute name="xsi:noNamespaceSchemaLocation">
        <xsl:text>
          ../Schema/BattlespaceInformationExchangeEnumeratedDomains.xsd
        </xsl:text>
      </xsl:attribute>
     <xsl:comment>Battlespace Information Exchange Entity definitions.
Extracted using XSLT from OpenOffice XML version of ANNEX E.  DATA MODEL
DOCUMENTATION—SPECIFICATIONS OF ENUMERATED DOMAINS from the Army Tactical
Command and Control Information System (ATCCIS) Working Group Working Paper 5-
5, Edition 5.0: The Land Command and Control Information Exchange Data Model
(LC2IEDM), ATTCIS Baseline 2.0, 18 March 2002.</xsl:comment>
     <xsl:apply-templates select="$AnnexE"/>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <!-- There are multiple tables in Annex E, each one describing an enumerated
Domain-->
  <xsl:template match="table:table">
    <!--Create 'Domain' Element-->
    <xsl:element name="Domain">
     <!--name is in first row, second cell-->
     <xsl:attribute name="name">
      <xsl:value-of select="normalize-space(table:table-row[1]/table:table-
cell[2])"/>
     </xsl:attribute>
     <!--definition is in second row, second cell-->
     <xsl:attribute name="definition">
      <xsl:value-of select="normalize-space(table:table-row[2]/table:table-
cell[2])"/>
     </xsl:attribute>
     <!--source is in third row, second cell-->
     <xsl:attribute name="source">
      <xsl:value-of select="normalize-space(table:table-row[3]/table:table-
cell[2])"/>
     </xsl:attribute>
     <!-- The 5th row has child data that is extracted using another template--
>
     <xsl:apply-templates select="table:table-row[5]"/>
<!--The row after the row whose first cell has the text 'USAGE' in it also
needs to be extracted using another template.  We have to do it this way
because we don't know how many rows precede this one-->
```

```
      <xsl:apply-templates select="table:table-row[preceding-
sibling::table:table-row/table:table-cell/text:p='USAGE']"/>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->

  <xsl:template match="table:table-row">
    <xsl:for-each select="following-sibling::table:table-row[following-
sibling::table:table-row/table:table-cell/text:p='USAGE']">
      <!--Create the 'Value' Element-->
      <xsl:element name="Value">
        <!--name is in cell #1-->
        <xsl:attribute name="name">
          <xsl:value-of select="normalize-space(table:table-cell[1])"/>
        </xsl:attribute>
        <!--definition is in cell #2-->
        <xsl:attribute name="definition">
          <xsl:value-of select="normalize-space(table:table-cell[2])"/>
        </xsl:attribute>
        <!--source is in cell #3-->
        <xsl:attribute name="source">
          <xsl:value-of select="normalize-space(table:table-cell[3])"/>
        </xsl:attribute>
        <!--physicalValue is in cell #4-->
        <xsl:attribute name="physicalValue">
          <xsl:value-of select="normalize-space(table:table-cell[4])"/>
        </xsl:attribute>
        <!--identifier is in cell #5-->
        <xsl:attribute name="identifier">
          <xsl:value-of select="normalize-space(table:table-cell[5])"/>
        </xsl:attribute>
      </xsl:element>
    </xsl:for-each>
  </xsl:template>
  <xsl:template match="table:table-row[preceding-sibling::table:table-
row/table:table-cell/text:p='USAGE']">
<!--Parse the text for each text:p element in the first cell of the row-->
    <xsl:for-each select="following-sibling::table:table-row/table:table-
cell[1]/text:p/text()">
      <!--Remember the position so as to match it with the value in the next
cell in the same position -->
      <xsl:variable name="pos" select="position()"/>
      <!--Create the 'Usage' Element-->
      <xsl:element name="Usage">
        <!--the entity attribute is the text:p in the first cell of this row that
has the same position-->
        <xsl:attribute name="entity">
          <xsl:value-of select="normalize-space(ancestor::table:table-
row/table:table-cell[1]/text:p/text()[$pos])"/>
        </xsl:attribute>
        <!--the attribute is the text:p in the second cell of this row that has
the same position-->
        <xsl:attribute name="attribute">
          <xsl:value-of select="normalize-space(ancestor::table:table-
row/table:table-cell[2]/text:p/text()[$pos])"/>
        </xsl:attribute>
        <!--the optionality attribute is the text:p in the third cell of this row
that has the same position-->
        <xsl:attribute name="optionality">
          <xsl:value-of select="normalize-space(ancestor::table:table-
row/table:table-cell[3]/text:p/text()[$pos])"/>
        </xsl:attribute>
      </xsl:element>
```

```
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

9. MakeSchema.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- JD Neushul, Naval Postgraduate School-->
<xsl:stylesheet version="1.1" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="yes"/>
  <!-- Remove whitespace -->
  <xsl:strip-space elements="*"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <!--In order to access the XML documents that contain the data, the
'document()' function must be used.  This powerful function allows access to
XML documents on the local computer, or anywhere on the Web.  in this case and
indexing convention is used by putting the locations of the files in the same
XML file that has the location of the content data.  This way when the
locations of these files change, it will only be necessary to change the URLs
in one file.  -->
<xsl:param name="SourceDocPath" select="document('ContentPath.xml')/Paths"/>
  <xsl:variable name="EntityRelations"
select="document($SourceDocPath/@EntityRelationships)"/>
  <xsl:variable name="EntityDefs"
select="document($SourceDocPath/@EntityDefinitions)"/>
  <xsl:variable name="AttributeDefs"
select="document($SourceDocPath/@AttributeDefinitions)"/>
  <xsl:variable name="AttributeValues"
select="document($SourceDocPath/@EnumeratedDomains)"/>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="/">
    <xsl:element name="xs:schema">
      <!--This XSLT produces an XML Schema, which is an XML document that is
compliant with the W3C XML Schema Specification, which is itself expressed as
an XML Schema and is used as a namespace. -->
      <xsl:attribute name="xmlns:xs">
        <xsl:text>http://www.w3.org/2001/XMLSchema</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="elementFormDefault">
        <xsl:text>qualified</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="attributeFormDefault">
        <xsl:text>qualified</xsl:text>
      </xsl:attribute>
      <!-- Begin Root Element of Schema-->
      <xsl:element name="xs:element">
        <xsl:attribute name="name">
          <xsl:text>BattlespaceData</xsl:text>
        </xsl:attribute>
        <!--  Parent Entities and Child Entities are described in the
EntityRelationships.xml document, from Annex C-->
        <xsl:element name="xs:complexType">
          <xsl:element name="xs:sequence">
            <!--Apply a reference to each ParentEntity Element that is a Key
Entity.  This results in a top level that matches Figure 3-2 Key Entities of
the Generic Hub Data Model on page 21 of the C2IEDM Specification-->
            <xsl:apply-templates
select="$EntityRelations/*/ParentEntity[@name='ACTION' or @name='CANDIDATE-
TARGET-LIST' or @name='CAPABILITY' or @name='CONTEXT' or @name='LOCATION' or
@name='OBJECT-ITEM' or @name='OBJECT-TYPE' or @name='REPORTING-DATA' or
@name='RULE-OF-ENGAGEMENT']"/>
          </xsl:element>
```

```
        </xsl:element>
      </xsl:element>
      <!--Apply a template to each ParentEntity Element to make them all Global
Elements.  this means they are at the top level of the Schema Tree-->
      <xsl:apply-templates select="$EntityRelations/*/ParentEntity"
mode="Global"/>
      <!--Apply a template to each EnumeratedDomains Element to make them all
Global Elements.  this means they are at the top level of the Schema Tree-->
      <xsl:apply-templates select="$AttributeValues/*/Domain"/>
      <!--End Root Element-->
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <!--ParentEntity Template.  Parent Entities occur at many different levels in
the tree.  The Independent Entities only appear at the top level.  Parent
Entities are designated as Global Elements and are Referenced vice repeated
after they first occur.  This means that they all have to be at the top level
of the Schema, but will be referenced in the tree structure-->
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="ParentEntity" mode="Global">
    <xsl:variable name="ParentName" select="@name"/>
    <xsl:element name="xs:element">
      <xsl:attribute name="name">
        <xsl:value-of select="@name"/>
      </xsl:attribute>
      <!-- Annotations are extremely important to maintain in a Schema that is
this large and complex.-->
      <xsl:element name="xs:annotation">
        <xsl:element name="xs:documentation">
          <xsl:value-of
select="$EntityDefs/*/Entity[@name=$ParentName]/@definition"/>
        </xsl:element>
      </xsl:element>
      <!-- Now that a ParentEntity has been created, it's children can be
discovered.-->
      <xsl:element name="xs:complexType">
        <xsl:element name="xs:sequence">
          <xsl:apply-templates
select="$EntityDefs/*/Entity[@name=$ParentName]/Attribute"/>
          <xsl:apply-templates select="ChildEntity"/>
        </xsl:element>
        <xsl:element name="xs:attribute">
          <xsl:attribute name="name">
            <xsl:text>owner_id</xsl:text>
          </xsl:attribute>
          <xsl:attribute name="use">
            <xsl:text>required</xsl:text>
          </xsl:attribute>
          <xsl:element name="xs:annotation">
            <xsl:element name="xs:documentation">
              <xsl:text>The unique value, assigned to represent a specific
proprietor of a certain data item (record) that is responsible for maintaining
that data item.</xsl:text>
            </xsl:element>
          </xsl:element>
        </xsl:element>
        <xsl:element name="xs:attribute">
          <xsl:attribute name="name">
            <xsl:text>update_seqnr</xsl:text>
          </xsl:attribute>
          <xsl:attribute name="use">
            <xsl:text>required</xsl:text>
          </xsl:attribute>
```

```
          <xsl:element name="xs:annotation">
            <xsl:element name="xs:documentation">
              <xsl:text>An absolute sequence number, assigned to represent the
validity (in terms of seniority) of a certain data item.</xsl:text>
            </xsl:element>
          </xsl:element>
        </xsl:element>
      </xsl:element>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <!--This adds a ParentEntity as a reference wherever it occurs in the tree.-->
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="ParentEntity">
    <xsl:element name="xs:element">
      <xsl:attribute name="ref">
        <xsl:value-of select="@name"/>
      </xsl:attribute>
      <!-- There can be multiple ParentEntity nodes in a BGH data entry-->
      <xsl:attribute name="maxOccurs">
        <xsl:text>unbounded</xsl:text>
      </xsl:attribute>
      <!-- If this is a Key Entity then it is optional .. This makes all of the
Key Entities optional.  A valid BGH data entry doesn't require all or any of
them.-->
      <xsl:if test="@name='ACTION' or @name='CANDIDATE-TARGET-LIST' or
@name='CAPABILITY' or @name='CONTEXT' or @name='LOCATION' or @name='OBJECT-
ITEM' or @name='OBJECT-TYPE' or @name='REPORTING-DATA' or @name='RULE-OF-
ENGAGEMENT'">
        <xsl:attribute name="minOccurs">
          <xsl:text>0</xsl:text>
        </xsl:attribute>
      </xsl:if>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <!-- If the Attribute name is in the EnumerationDomain - (all Globalized) -
Make it a Reference -->
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="Attribute[@name=$AttributeValues/*/Domain/@name]">
    <xsl:variable name="AttName" select="@name"/>
    <xsl:element name="xs:element">
      <xsl:attribute name="ref">
        <xsl:value-of select="@name"/>
      </xsl:attribute>
      <xsl:if
test="$AttributeValues/*/Domain[@name=$AttName]/Usage/@optionality='OP'">
        <xsl:attribute name="minOccurs">
          <xsl:text>0</xsl:text>
        </xsl:attribute>
      </xsl:if>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <!-- Non-global Attribute    -->
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="Attribute">
    <xsl:variable name="AttName" select="@name"/>
    <xsl:element name="xs:element">
      <xsl:attribute name="name">
        <xsl:value-of select="@name"/>
      </xsl:attribute>
      <xsl:element name="xs:annotation">
```

```
        <xsl:element name="xs:documentation">
          <xsl:value-of
select="$AttributeDefs/*/Attribute[@name=$AttName]/@definition"/>
          <xsl:apply-templates
select="$AttributeValues/*/Domain[@name=$AttName]/@source"/>
        </xsl:element>
      </xsl:element>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <!-- Enumeration Domains -->
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template match="Domain">
    <xsl:variable name="DomName" select="@name"/>
    <xsl:element name="xs:element">
      <xsl:attribute name="name">
        <xsl:value-of select="@name"/>
      </xsl:attribute>
      <xsl:element name="xs:annotation">
        <xsl:element name="xs:documentation">
          <xsl:value-of
select="$AttributeDefs/*/Attribute[@name=$DomName]/@definition"/>
          <xsl:apply-templates
select="$AttributeValues/*/Domain[@name=$DomName]/@source"/>
        </xsl:element>
      </xsl:element>
      <xsl:if test="count($AttributeValues/*/Domain[@name=$DomName]/Value)">
        <xsl:element name="xs:complexType">
          <xsl:if test="string-length(@key)&gt;0">
            <xsl:apply-templates select="@key"/>
          </xsl:if>
          <xsl:element name="xs:choice">
            <xsl:apply-templates
select="$AttributeValues/*/Domain[@name=$DomName]/Value"/>
          </xsl:element>
        </xsl:element>
      </xsl:if>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <!-- If ChildEntity is also a ParentEntity, just make a Reference, and make
optional   -->
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
  <xsl:template
match="ChildEntity[@name=$EntityRelations/*/ParentEntity/@name]">
    <xsl:variable name="EntityName" select="@name"/>
    <!--Prevent Duplicates.-->
    <xsl:if test="not(preceding-sibling::*/@name=$EntityName)">
      <xsl:element name="xs:element">
        <xsl:attribute name="ref">
          <xsl:value-of select="@name"/>
        </xsl:attribute>
        <xsl:attribute name="minOccurs">
          <xsl:text>0</xsl:text>
        </xsl:attribute>
        <xsl:if test="contains(@cardinality,'-More')">
          <xsl:attribute name="maxOccurs">
            <xsl:text>unbounded</xsl:text>
          </xsl:attribute>
        </xsl:if>
      </xsl:element>
    </xsl:if>
  </xsl:template>
```

```
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<!-- If ChildEntity not a ParentEntity as well - No Global Reference -->
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="ChildEntity">
  <xsl:variable name="EntityName" select="@name"/>
  <!--Prevent Duplicates.-->
  <xsl:if test="not(preceding-sibling::*/@name=$EntityName)">
    <xsl:element name="xs:element">
      <xsl:attribute name="name">
        <xsl:value-of select="@name"/>
      </xsl:attribute>
      <xsl:if test="contains(@cardinality,'to-Zero')">
        <xsl:attribute name="minOccurs">
          <xsl:text>0</xsl:text>
        </xsl:attribute>
      </xsl:if>
      <xsl:if test="contains(@cardinality,'-More')">
        <xsl:attribute name="maxOccurs">
          <xsl:text>unbounded</xsl:text>
        </xsl:attribute>
      </xsl:if>
      <xsl:element name="xs:annotation">
        <xsl:element name="xs:documentation">
          <xsl:value-of
select="$EntityDefs/*/Entity[@name=$EntityName]/@definition"/>
        </xsl:element>
      </xsl:element>
      <xsl:element name="xs:complexType">
        <xsl:element name="xs:sequence">
          <xsl:apply-templates
select="$EntityDefs/*/Entity[@name=$EntityName]/Attribute"/>
          <xsl:apply-templates
select="$EntityRelations/*/ParentEntity[@name=$EntityName]/*"/>
        </xsl:element>
        <xsl:apply-templates select="@cardinality"/>
        <xsl:apply-templates select="LogicalForeignKey"/>
        <xsl:apply-templates select="@relationship"/>
        <xsl:apply-templates select="@verbPhrase"/>
      </xsl:element>
    </xsl:element>
  </xsl:if>
</xsl:template>
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<!-- Adds Values from EnumerationDomain -->
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="Value">
  <xsl:variable name="ValName" select="parent::*/@name"/>
  <xsl:element name="xs:element">
    <xsl:attribute name="name">
      <xsl:choose>
        <!-- Element names can't start with numbers.. -->
        <xsl:when test="number(substring(@physicalValue,1,1))">
          <xsl:value-of select="concat($ValName,@physicalValue)"/>
        </xsl:when>
        <xsl:when test="substring(@physicalValue,1,1)='0'">
          <xsl:value-of select="concat($ValName,@physicalValue)"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of select="@physicalValue"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
```

```
      <xsl:if
test="$AttributeValues/*/Domain[@name=$ValName]/Usage/@optionality='OP'">
        <xsl:attribute name="minOccurs">
          <xsl:text>0</xsl:text>
        </xsl:attribute>
      </xsl:if>

      <xsl:element name="xs:annotation">
        <xsl:element name="xs:documentation">
          <xsl:apply-templates select="@source"/>
          <xsl:value-of select="@definition"/>
        </xsl:element>
      </xsl:element>

      <xsl:element name="xs:complexType">
        <xsl:apply-templates select="@name"/>
        <xsl:apply-templates select="@identifier"/>
      </xsl:element>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="@identifier">
    <xsl:element name="xs:attribute">
      <xsl:attribute name="name">
        <xsl:text>id</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="use">
        <xsl:text>required</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="fixed">
        <xsl:value-of select="."/>
      </xsl:attribute>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="@name">
    <xsl:element name="xs:attribute">
      <xsl:attribute name="name">
        <xsl:text>value</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="use">
        <xsl:text>required</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="fixed">
        <xsl:value-of select="."/>
      </xsl:attribute>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
  <xsl:template match="@key">
    <xsl:element name="xs:attribute">
      <xsl:attribute name="name">
        <xsl:text>key</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="use">
        <xsl:text>optional</xsl:text>
      </xsl:attribute>
      <xsl:attribute name="fixed">
        <xsl:value-of select="."/>
      </xsl:attribute>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_*-->
```

```
<xsl:template match="@source">
  <xsl:attribute name="source">
    <xsl:value-of select="."/>
  </xsl:attribute>
</xsl:template>
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="@relationship">
  <xsl:element name="xs:attribute">
    <xsl:attribute name="name">
      <xsl:text>relation</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="use">
      <xsl:text>optional</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="fixed">
      <xsl:value-of select="."/>
    </xsl:attribute>
  </xsl:element>
</xsl:template>
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="@verbPhrase">
  <xsl:element name="xs:attribute">
    <xsl:attribute name="name">
      <xsl:text>verbPhrase</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="use">
      <xsl:text>optional</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="fixed">
      <xsl:value-of select="."/>
    </xsl:attribute>
  </xsl:element>
</xsl:template>
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="@cardinality">
  <xsl:element name="xs:attribute">
    <xsl:attribute name="name">
      <xsl:text>cardinality</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="use">
      <xsl:text>optional</xsl:text>
    </xsl:attribute>
    <xsl:attribute name="fixed">
      <xsl:value-of select="."/>
    </xsl:attribute>
  </xsl:element>
</xsl:template>
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="LogicalForeignKey">
  <xsl:element name="xs:attribute">
    <xsl:attribute name="name">
      <xsl:choose>
        <xsl:when test="preceding-sibling::LogicalForeignKey">
          <xsl:value-of select="concat('foreignKey',position())"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>foreignKey</xsl:text>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <xsl:attribute name="use">
      <xsl:text>optional</xsl:text>
    </xsl:attribute>
```

```
      <xsl:attribute name="fixed">
        <xsl:value-of select="."/>
      </xsl:attribute>
    </xsl:element>
  </xsl:template>
  <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
</xsl:stylesheet>
```

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

BOOKS

Ames, A., Nadeau, D. and Moreland, J., *VRML 2.0 sourcebook (2nd ed.)*, New York: John Wiley and Sons, 1997

Cagle, Kurt and Others, *Professional XSL*, Wrox Press, 2001

Christensen, Clayton M., *The Innovators Dilemma, When New Technologies Cause Great Firms to Fail*, Harvard Business School Press, Boston Massachusetts, 1997

Ferber, Jacques, *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999

Graves, Mark, *Designing XML Databases*, Prentice Hall, 2002

Hall, Marty, *More Servlets and JavaServer Pages*, Sun Microsystems Press/Prentice Hall, 2002

Harold, Elliotte Rusty, *Processing XML with Java*, Addison-Wesley, 2002

Harold, Elliotte Rusty , *The XML Bible, Gold Edition*, Hungry Minds, 2001

Hunter, David, Cagle, Kurt and Dix, Chris, *Beginning XML, 2nd Edition*, Wrox Press, 2001

Kay, Michael, *XSLT Programmer's Reference 2nd Edition*, Wrox Press, 2001

Mahmoud, Qusay H., *Distributed Programming with Java*, Manning, 2000

White, Chuck, Quin, Liam and Burman, Linda, *Mastering XML, Premium Edition*, SYBEX, 2001

Wooldridge, Michael , *An Introduction to MultiAgent Systems*, Wiley, 2002

PAPERS

Abernathy, M. and Shaw, S., *Integrating Geographic Information in VRML Models*, Proceedings of VRML Symposium, 1998

Becker , David, *Taking XML's measure*, CNET News.com, ZDNN, http://www.zdnn.com. 2002 September 24, 2003, Accessed Sept 2003

Berners-Lee, Tim, Hendler, James and Lassila, Ora, , Scientific American, May 2001

Blais, Curtis, Brutzman, Don, Horner, Doug and Nicklaus, Shane, *Web Based 3D Technology for Scenario Authoring and Visualization: The Savage Project*, Proceedings of Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), 2001

Brutzman, Don, *The Virtual Reality Modeling Language and Java*, Communications of the ACM vol. 41 no. 6, pp. 57-64, June 1998

Brutzman, Don, *Graphics Internetworking: Bottlenecks and Breakthroughs*, chapter four, Digital Illusions, Clark Dodsworth editor, Addison-Wesley, Reading Massachusetts, August 1997

Burke, Erik M., *XSLT Processing with Java, Related Reading*, The O'Reilly Network (http://www.oreillynet.com/), 2002 Accessed Sept 2003

Carey, Scott A., Kleiner, Martin S., Hieb, Michael R. and Brown, Richard, *Standardizing Battle Management Language; A Vital Move Towards the Army Transformation*, Proceeding of the 2d Battle Management Language Symposium, 21-22 August 02

Carey, Scott A., Kleiner, Martin S., Hieb, Michael R. and Brown, Richard, *Standardizing Battle Management Language; Facilitating Coalition Interoperability*, Proceeding of the 2d Battle Management Language Symposium, 21-22 August 02

Carr, Francis H. and Hieb, Michael R., *M&S Interoperability within the DII COE: Building a Technical Requirements Specification*, Paper 00F-SIW-133 at the Spring Simulation Interoperability Workshop 2001, Orlando, Florida, March 2001

Ceruti, M., Anken, C., Lin , A. and Rubin, S., *Applications of High-Performance Knowledge-Based Technology*, Proceedings of IEEE Systems Man and Cybernetics, 2000

Ceruti, M.G., *Application of Knowledge-Base Technology for Problem Solving in Information-Systems Integration*, Proceedings of the DOD Database Colloquium 97, pp. 215-234, September 1997

Ceruti, M.G., *Establishing a DataMining Environment for Wartime Event Prediction with an Object-Oriented Command and Control Database*, Proceedings of the IEEE International Symposium on Object-oriented Real-time Distributed Computing, ISORC2K, March 2000

Ceruti, M.G., *Mobile Agents in Network-Centric Warfare*, IEICE Transactions on Communications, Vol.E84-B No.10, October 2001

Ceruti, M.G., *An Expanded Review of Information-System Terminology*, Proceedings of AFCEAs Sixteenth Annual Federal Database Colloquium and Exposition, Information Dominance and Assurance, San Diego, California, September 1999

Ceruti, M.G., *Challenges in Data Management for the United States Department of Defense (DoD) Command, Control, Communications, Computers, and Intelligence (C4I) Systems*, Proceedings of the Twenty-Second Annual International IEEE Computer Software and Applications Conference, Vienna, Austria, pp 622-629, August 1998

Ceruti, M.G. and Kamel, Magdi N., *Semantic Heterogeneity in Database and Data Dictionary Integration for Command and Control Systems*, Proceedings of the Eleventh Annual DOD Database Colloquium, Database Integration and Innovation, AFCEA, pp 65-89, 29-31 August, 1994

Chaum, Erik, *Achieving the Joint Vision: The Role of Operational Context in Future Systems*, U S Naval Undersea Warfare Center Division, Newport, 2002

Chen, Peter Pin-Shan Chen, *The Entity Relationship Model,Toward a Unified View of Data*, ACM Transactions on Database Systems, pp. 9 - 36, March 1976

Daly, John J. and Tolk, Andreas, *Modeling and Simulation Integration with Network-Centric Command and Control Architectures*, 2003 Fall Simulation Interoperability Workshop, Orlando, Florida, September 2003

Duce, David, Hopgood, Bob and Lightfoot, David, *SVG and X3d: Technology for a Foundation Course*, Eurographics/ACM SIGGRAPH Workshop on Computer Graphics Education, Support for Computer Graphics Educators, Bristol University, UK, July 6-7 2002

Fishwick, Paul A., *Using XML For Simulation Modeling*, Proceedings of the 2002 Winter Simulation Conference, San Diego, California, December 8-11 2002

Fishwick, Paul A., *The rube Framework for Software Modeling and Customized 3-D Visualization*, Journal of Visual Languages and Computing, July 2002

Ford, Paul, *Marking Up Bureaucracy*, Published on XML.com http://www.xml.com/pub/a/2003/09/24/government.html, O'Reilly and Associates, Inc., 2003

Goodwin, Cecil W. H., Gordon, Stephen R. and Siegel, David, *Re-interpreting The Location Referencing Problem: A Protocol Approach*, Proceedings of GIS-T 95 Symposium, Washington DC: American Association of State Highway and Transportation Officials (AASHTO), 1995

Grasmeyer, Joel M. and Keennon, Matthew T., *Development of the Black Widow Micro Air Vehicle*, American Institute of Aeronautics and Astronautics, AIAA-2001-0127, 2001

Hartel, Robert and Chamberlain, Sam, *Multinational Interoperability Requirements, A Core Competency*, 5th International Command and Control Research and Technology Symposium Australia War Memorial, Canberra, ACT, Australia, 24-26 October 2000

Hieb, Michael R. and Blalock, James, *Data Alignment between Army C4I Databases and Army Simulations*, 1999 Spring Simulation Interoperability Workshop, Orlando, Florida, March 2001

Kogut, Paul, Cranefield, Stephen, Hart, Lewis, Dutra, Mark, Baclawski, Kenneth, Kokar, Mieczyslaw and Smith, Jeffrey, *UML for Ontology Development*, Knowledge Engineering Review Journal Special Issue on Ontologies in Agent Systems, Vol. 17, 2002

Krusche, Stefan and Tolk, Andreas, *A SHADE Approach for Coupling C4I-Systems and Simulation Systems*, 1999 Fall Simulation Interoperability Workshop, September 1999

Krusche, Stefan and Tolk, Andreas, *Information Processing as a Key Factor for Modern Federations of Combat Information Systems*, NATO/RTO Information Systems Technology Panel (IST) Symposium "New Information Processing Techniques for Military Systems", Istanbul, October 2000

Macedonia, M.R. and Zyda , M.J., *A Taxonomy for Networked Virtual Environments*, IEEE Multimedia, Vol. 4, No. 1, 1997

Mayk, Israel and Klose, Dirk, *Multinational C2 Experiments Supported by C2 Systems and Modeling and Simulations Addressing Army Transformation of Collaborative Planning and Interoperable Execution in a Coalition Environment*, Paper for the 8th International Command and Control Research and Technology Symposium, May 2003

Moore, Robert, Dowding, John, Bratt, Harry, Gawron, J. M., Gorfu, Yonael and Cheyer, Adam, *CommandTalk: A Spoken-Language Interface for Battlefield Simulations*, Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington, DC pp. 1-7, Association for Computational Linguistics., 1997

Nord, James, Synnes, K°are and Parnes, Peter, *An Architecture for Location Aware Applications*, 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9, January 2002

Reddy, M., Leclerc, Y. G. and Heller, A., *GeoVRML: Open Web-based 3D Cartography*, Proceedings of the International Cartographic Conference (ICC2001), Beijing, 6-10 August 2001

Reddy, M., Iverson, L. and Leclerc, Y. G., *GeoVRML 1.0: Adding Geographic Support to VRML*, GeoInformatics, Volume 3, September 2000

Reddy, M., Leclerc, Y. G., Iverson, L. and Bletter, N., *TerraVision II: Visualizing Massive Terrain Databases in VRML*, IEEE Computer Graphics and Applications (Special Issue on VRML), 19(2): 30-38., 1999

Reddy, M., Iverson, L. and Leclerc, Y. G., *Enabling Geographic Support in Virtual Reality Modeling with GeoVRML*, Cartography and Geographic Information Science. 26(3):180, 1999

Rhyne, Theresa-Marie, Brutzman, Don and Macedonia, Michael , *Internetworked Graphics and the Web*, IEEE COMPUTER, vol. 30 no. 8, pp.99-101, August 1997

Rhyne, T. M. , *Going Virtual with Geographic Information and Scientific Visualization*, 1997

Rosenthal, Arnon, Seligman, Len and Costello, Roger, *XML, Databases, and Interoperability*, Federal Database Colloquium, AFCEA, San Diego, 1999

Shapira, Andrew, *Fast Line-Edge Intersections on a Uniform Grid*, ``Graphics Gems,'' Andrew Glassner Ed., Academic Press Inc., 1990.

Tolk, Andreas, *Bridging the Data Gap. Recommendations for Short, Medium, and Long Term Solutions*, Paper 01S-SIW-011 at the Spring Simulation Interoperability Workshop 2001, Orlando, Florida, March 2001

Tolk, Andreas, *A Common Framework for Military M&S and C4I Systems*, 2003 Spring Simulation Interoperability Workshop, April 2003

Tolk, Andreas, *Using ATCCIS as an Information Layer to couple CGF Federates and Closed Combat Simulations*, 1999 Fall Simulation Interoperability Workshop, Orlando, Florida, September 1999

Tolk, Andreas and Hieb, Michael R., *Building and Integrating M&S Components into C4ISR Systems for Supporting Future Military Operations*, 2003 International Conference on Grand Challenges for Modeling and Simulation (ICGCMS ’03), August, 2002

Tolk, Andreas and Sinclair, Mark R., *Building up a Common Data Infrastructure*, NATO Symposium on the "Analysis of the Military Effectiveness of Future C2 Concepts and Systems" organized by the Studies, Analyses and Simulation Panel (SAS), NC3 Agency, The Hague, The Netherlands, April 2002

Tolk, Andreas, *New Overarching Approaches to Improve IT/C4ISR Systems by Integrating M&S Components*, Computers and Geosciences, 23(4): 489-491., December 2002

Tolk, Andreas, *Model Scheme A Good Fit for C4ISR*, COTS Journal [87], October 2002

Tso, Kam S., Tharp, Gregory K., Zhang, Wayne and Tai, Ann T., *Multi-Agent Operator Interface for Unmanned Aerial Vehicles*, The 18th Digital Avionics Systems Conference St. Louis, MO, October 24-29, 1999

Wilcox , Joe, *Microsoft limits XML in Office 2003*, CNET News.com, ZDNN, http://www.zdnn.com., April 11, 2003

Wilson, J.R., *UAVS and the Human Factor*, Aerospace America Magazine, June 2002

Zyda, Michael J., Pratt, David, Falby, John, Lombardo, Chuck and Kelleher, Kristen, *The Software Required for the Computer Generation of Virtual Environments*, Presence, 2, 2, March 1993

Zyda, Michael J., Macedonia , M.R., Pratt, David, Brutzman, Donald P. and Barham, Paul T., *Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments*, IEEE Computer Graphics Applications, September 1995

Zyda, Michael J., Lombardo, Chuck and Pratt, David, *Hypermedia and Networking in the Development of Large-Scale Virtual Environments*, The Proceedings of the International Conference on Artificial Reality and Tele-existence, Tokyo, Japan, 6-8 July 1993

SPECIFICATIONS

Department of Defense, *MIL-PRF-89020B Digital Terrain Elevation Data (DTED)*, 27 July 2001

Department of Defense, *MIL-STD 89045 Performance Specification for Geospatial Symbols for Digital Displays*, 20 February 1998

Open GIS Consortium, Inc., *OGC 00-029, Geography Markup Language (GML) v1.0*, May 2000

Open GIS Consortium, Inc., *OGC 02-023r4, OpenGIS Geography Markup Language (GML) Implementation Specification*, January 2003

Open GIS Consortium, Inc., *OGC 01-047r1, Web Map Service Implementation Specification*, January 2001

Web3D Consortium, *GeoVRML 1.1 Specification*, 2001

Web3D Consortium, *ISO/IEC 19775:200x, X3D, Information technology, Computer graphics and image processing, Extensible 3D (X3D)*, 2001

Web3D Consortium, *ISO/IEC 19776:200x, X3D Encodings, Information technology, Computer graphics and image processing, Extensible (X3D) Encodings*, 2001

Web3D Consortium, *ISO/IEC 19776:200x, X3D Encodings, Information technology, Computer graphics and image processing, Extensible (X3D) Encodings*, 2001

World Wide Web Consortium, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation, October 2000

World Wide Web Consortium, *Extensible Markup Language (XML) 1.1*, W3C Candidate Recommendation, October 2002

World Wide Web Consortium, *XML Inclusions (XInclude) version 1.0*, W3C Candidate Recommendation, September 2002

World Wide Web Consortium, *Namespaces in XML 1.1*, W3C Candidate Recommendation, December 2002

World Wide Web Consortium, *XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)*, W3C Recommendation, October 2002

World Wide Web Consortium, *XML Schema Part 0: Primer*, W3C Recommendation, May 2001

World Wide Web Consortium, *XML Schema Part 1: Structures*, W3C Recommendation, May 2001

World Wide Web Consortium, *XML Schema Part 2: Datatypes*, W3C Recommendation, May 2001

World Wide Web Consortium, *XML Schema: Formal Description*, W3C Recommendation, March 2001

World Wide Web Consortium, *XML Schema Requirements*, W3C Recommendation, February 1999

World Wide Web Consortium, *XSL Transformations (XSLT)*, W3C Recommendation, November 1999

World Wide Web Consortium, *XSL Transformations (XSLT) version 2.0*, W3C Working Draft, May 2003

World Wide Web Consortium, *XML Path Language (XPath) version 1.0*, W3C Recommendation, November 1999

World Wide Web Consortium, *Scalable Vector Graphics (SVG) 1.0 Specification*, W3C Recommendation, September 2001

World Wide Web Consortium, *OWL Web Ontology Language Overview*, Candidate Recommendation, 18 August 2003

PUBLIC DOCUMENTS

Address to Joint Battle Management Command and Control Summit by Michael Wynne, Acting Under Secretary of Defense, *Joint Battle Management Command and Control; Transforming the Battlespace*, July 30, 2003

Chairman of the Joint Chiefs of Staff, *Instruction (CJCSI 6212.01B), Interoperability and Supportability of Natiional Security Systems, and Information Technology Systems*, 8 May 2000

Defense Modeling and Simulation Office, *DoD 5000.59-M DoD Modeling and Simulation (M&S) Glossary*, January 1998

Defense Modeling and Simulation Office, *DoD 5000.59-P Modeling and Simulation (M&S) Master Plan*, October 1995

Department of Defense, *Joint Publication (Joint Pub) 1-02, "Department of Defense Dictionary of Military and Associated Terms"*, April 2001(As Amended Through June 2003)

Headquarters United States Marine Corps, *Marine Corps Capstone Concept, "Expeditionary Maneuver Warfare"*, November 2001

Headquarters United States Marine Corps, *Marine Corps Doctrinal Publication (MCDP) 1, "Marine Corps Operations"*, September 2001

Headquarters United States Marine Corps, *Marine Corps Doctrinal Publication (MCDP) 1-0, "Warfighting"*, June 1997

Headquarters United States Marine Corps, *Marine Corps Doctrinal Publication (MCDP) 5, "Planning"*, July 1997

Headquarters United States Marine Corps, *Marine Corps Reference Publication (MCRP) 5-12C, "Marine Corps Supplement to the DOD Dictionary of Military and Associated Terms"*, July 1998

Headquarters United States Marine Corps, *Marine Corps Strategy 21*, November 2000

Headquarters United States Marine Corps, *Marine Corps Warfighting Publication (MCWP) 5-1, "Marine Corps Planning Process"*, January 2000

Headquarters United States Marine Corps, *USMC Concepts and Programs 2003*, 2003

U.S. Department of Defense, *U.S. Department of Defense Directive (DODD) 8100.1: "Global Information Grid (GIG) Overarching Policy," The Pentagon, Washington, D.C.*, September 2002

U.S. Department of Defense Office of the Assistant Secretary of Defense (Command, Control, Communications, Intelligence) , *Web Site Administration Policies and Procedures*, 2002

U.S. Office of Management and Budget, *CIRCULAR NO. A-119,Federal Participation in the Development and Use of Voluntary Consensus Standards and in Conformity Assessment Activities*, February 10, 1998

United States 104th Congress, *Public Law 104-113, National Technology Transfer and Advancement Act of 1995*, 1995

TECHNICAL REPORTS

Department of Defense, *Network-Centric Warfare, Department of Defense Report to Congress*, 27 July 2001

Environmental Systems Research Inc (ESRI), *Architectural Solution and Standards Compliance*

Environmental Systems Research Inc (ESRI), *Functional Requirements Satisfaction*

Headquarters United States Marine Corps, *Marine Corps Concept Paper(MCCP) BEYOND C2; A Concept for Comprehensive Command and Coordination of the Marine Air-Ground Task Force*, 02 Jun 1998

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 17-03. Exploitation and Pursuit in Modern War Part I-Recognizing Opportunity*, 25 April 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 18-03. Unrestricted Warfare - New Tactics or New Technologies*, 2 May 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 20-03. Wireless Technology*, 16 May 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 25-03. Efficiency vs. Effectiveness I*, 20 June 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 27-03. Efficiency vs. Effectiveness II*, 11 June 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 30-03. Battlespace Geometry in Modern Ground Combat*, 1 August 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 31-03. Technology Governance*, 08 August 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 32-03. Operational Readiness*, 15 August 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 34-03. The Military Staff*, 29 August 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 36-03. Vetronics and Telematics*, 24 January 2003

Headquarters United States Marine Corps, Plans, Policies, and Operations Department, Strategic Initiatives Group, *War Room Report 4-03. Military Organization*, 24 January 2003

Multilateral Interoperability Programme, *Brief to 2003 DoD Standardization Symposium, LtCol Scott Hoffman, USMC, Joint Staff J6I*, 4 March 2003

National Imagery and Mapping Agency (NIMA), *Commercial Joint Mapping Toolkit (C/JMTK) Functional Requirements Document (FRD) (or C/JMTK FRD)*, 26 January 2001

NATO, Army Tactical Command and Control Information System (ATCCIS) Working Group, *Overview of the Land C2 Information Exchange Data Model (LC2IEDM), ATCCIS Baseline 2.0*, 18 March 2002

NATO, Army Tactical Command and Control Information System (ATCCIS) Working Group, *The Land C2 Information Exchange Data Model, Working Paper 5-5, Edition 5.0, ATCCIS Baseline 2.0*, 18 March 2002

NATO, Army Tactical Command and Control Information System (ATCCIS) Working Group, *The Land C2 Information Exchange Data Model, Working Paper 5-5, Edition 5.0, ATCCIS Baseline 2.0 - ANNEXES*, 18 March 2002

Office of Naval Research(ONR), *Broad Agency Announcements 02-024 Development and Demonstration of Intelligent Autonomy in Unmanned Vehicles*, September 2002

U.S. Department of Defense, *Department of Defense Net-Centric Data Strategy*, May 9, 2003

United States Marine Corps, *1MARDIV Operation Iraqi Freedom Lessons Learned*, 2003

United States Marine Corps, *1MARDIV Operation Iraqi Freedom Observations Brief*, 2003

United States Marine Corps, *1st FSSG Operation Iraqi Freedom After Action Report*, 18 August, 2003

United States Marine Corps, *Dragon Eye Concept of Operations Description*, 6 June 02

United States Marine Corps, *Marine Corps Systems Command Liaison Team, Field Report, Central Iraq*, 20 April to 25 April 2003

United States Marine Corps, *Regimental Combat Team 5 Operation Iraqi Freedom Logistics After Action Report*, 2003

THESES / DISSERTATIONS

Harney, James William, *Analyzing Anti-Terrorist Tactical Effectiveness of Picket Boats for Force Protection of Navy Ships Using X3D Graphics and Agent Based Simulation. Master's Thesis*, Naval Postgraduate School, Monterey, California, March 2003

Hunsberger, Michael G. , *3D Visualization of Tactical Communications For Planning and Operations Using Virtual Reality Modeling Language (VRML) and Extensible 3D (X3D). Master's Thesis*, Naval Postgraduate School, Monterey, California, June 2001

Hutton, Claude O., *3D Battlespace Vizualization Using Operational Planning Data. Master's Thesis*, Naval Postgraduate School, Monterey, California, March 2003

Nicklaus, Shane, *Scenario Authoring and Visualization for Advanced Graphical Environments. Master's Thesis*, Naval Postgraduate School, Monterey, California, 2001

Reddy, Martin, *Perceptually Modulated Level of Detail for Virtual Environments. PhD Thesis.*, University of Edinburgh, 1997

Serin, Ekrim, *Design and Test of the Cross-Format Schema Protocol (XFSP) For Networked Virtual Environments. Master's Thesis.*, Naval Postgraduate School, Monterey, California, March 2003

WEB RESOURCES

Commercial Joint Mapping Toolkit (CJTK), *http://www.cjmtk.org*, Accessed: September 2003

Defense Modeling and Simulation Office, High Level Architecture, *https://www.dmso.mil/public/transition/hla/*, Accessed: September 2003

Defense Modeling and Simulation Office, High Level Architecture, *https://www.dmso.mil/public/transition/hla/*, Accessed: September 2003

Extensible Modeling and Simulation Framework (XMSF), *http://www.movesinstitute.org/xmsf/xmsf.html*, Accessed: September 2003

GeoVRML.org, *http://www.GeoVRML.org*, Accessed: September 2003

GOOGLE Search Engine, *http://www.google.com*, Accessed: September 2003

International Standards Organization(ISO), *http://www.iso.ch/iso/en/ISOOnline.frontpage*, Accessed: September 2003

Multilateral Interoperability Programme(MIP), *http://www.mip-site.org*, Accessed: September 2003

Naval Postgraduate School MOVES Institute, *http://www.movesinstitute.org/*, Accessed: September 2003

OpenOffice.org, The Native XML Office Suite, *http://www.OpenOffice.org*, Accessed: September 2003

Parallel Graphics Inc., *http://www.ParallelGraphics.com*, Accessed: September 2003

Scenario Authoring and Visualization for Advanced Graphical Environments - SAVAGE Library, *http://web.nps.navy.mil/~brutzman/Savage/contents.html*, Accessed: September 2003

SEDRIS, *http://www.sedris.org/*, Accessed: September 2003

The X3D Task Group, *http://www.web3d.org/x3d.html*, Accessed: September 2003

USJFCOM Distributed Continuous Experimentation Environment(DCEE), *http://www.jfcom.mil/about/fact_dcee.htm*, Accessed: September 2003

USJFCOM Joint Conflict and Tactical Simulation(JCATS), *http://www.jwfc.jfcom.mil/about/fact_jcats.htm*, Accessed: September 2003

Webopaedia.com, *http://www.pcwebopaedia.com/*, Accessed: September 2003

World Wide Web Consortium (W3C), *http://www.w3.org/*, Accessed: September 2003

XJ3D, *http://www.xj3d.org/*, Accessed: September 2003

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Brigadier General John R. Thomas, United States Marine Corps
   Director, Command, Control, Communications, and Computers (C4),
   HQ USMC, Department I
   Washington DC

4. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

5. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

6. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

7. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

8. Director, Studies and Analysis Division, MCCDC, Code C45
   Quantico, Virginia

9. Dr. Marion Ceruti
   SPAWAR
   San Diego, California

10. Dr. Peter Neushul
    University of California
    Santa Barbara, California

11. Associate Professor Don Brutzman
    Naval Postgraduate School
    Monterey, California

12. Research Associate Curt Blais
    Naval Postgraduate School
    Monterey, California

13. Research Associate Jeff Weekley
Naval Postgraduate School
Monterey, California

14. Captain James D. Neushul, USMC
Naval Postgraduate School
Monterey, California