2008-06

A merit-based architecture for the automatic selection and composition of services in soa-based C4ISR systems

Cook, Thomas S.

Monterey, California: Naval Postgraduate School, 2008.

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

A MERIT-BASED ARCHITECTURE FOR THE AUTOMATIC SELECTION AND COMPOSITION OF SERVICES IN SOA-BASED C4ISR SYSTEMS

by

Thomas S. Cook

June 2008

Dissertation Supervisor: James B. Michael

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2008	**3. REPORT TYPE AND DATES COVERED** Dissertation	
4. TITLE AND SUBTITLE: A Merit-Based Architecture for the Automatic Selection and Composition of Services in SOA-based C4ISR Systems		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Thomas S. Cook			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT **Approved for public release; distribution is unlimited**		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

Department of Defense (DoD) Command and Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems are responsible for supplying the right information at the right time to the warfighter. This dissertation presents a methodology for automating and realizing time-critical C4ISR applications. We introduce World Wide Web Consortium (W3C) compliant services into the planning and battle management processes where a computer can be more efficient and more effective than a human operator. We demonstrate our approach using ballistic missile defense (BMD) as a case study of a system in which the software services comprising the command, control, and battle management (C2BM) element of the BMD system need to operate within hard real-time constraints. We show the realization of time-critical C4ISR applications via continuously orchestrating individual services based on the automatically processing operational orders (OPORDs) and reports for the system to self-regulate itself. The system monitors, selects, and composes sub-services using a merit-based score until the mission stated in the OPORD is complete. The processing of the OPORDs for use by the C2BM element initiates and preserves the cyclic process of the kill chain used to negate threat ballistic missiles. To select and orchestrate services at runtime, we extended the current Web Services Description Language (WSDL) standard to encompass measures of performance (MOP) and measures of effectiveness (MOE). In our approach the WSDL-advertised measures are continuously updated based on runtime monitoring, creating an historical basis-of-confidence for each of the services. We demonstrate the generation and use by the C2BM of continuously updating service-selection criteria. Our composition language includes a software design pattern for use in ensuring time-critical processes complete within their time budget.

14. SUBJECT TERMS Service Oriented Architecture, Measures of Performance, Measures of Effectiveness, Quality of Service, C4ISR, Command and Control, Battle Management, software design, software architecture, architecture framework			**15. NUMBER OF PAGES** 177
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

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

A MERIT-BASED ARCHITECTURE FOR THE AUTOMATIC SELECTION AND COMPOSITION OF SERVICES IN SOA-BASED C4ISR SYSTEMS

Thomas S. Cook
Lieutenant Colonel, United States Army
B.S., Brockport Sate University, 1987
M.S., Naval Postgraduate School, 1999
M.S., University of Louisville, 2003

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN SOFTWARE ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2008

Author: _____
Thomas S. Cook

Approved by:

_____
James Bret Michael
Professor
Computer Science
and Electrical & Computer Engineering
Dissertation Supervisor

_____
Dan Boger
Professor, Dean of Research
Information Science

_____
Man-Tak Shing
Associate Professor
Computer Science

_____
Duminda Wijesekera
Associate Professor
Computer Science

_____
Doron Drusinsky
Associate Professor
Computer Science

_____
Dale “Butch” Caffall
Director, NASA
Independent Verification
and Validation Facility

Approved by: _____
Peter Denning, Chair, Department of Computer Science

Approved by: _____
Doug Moses, Associate Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Department of Defense (DoD) Command and Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems are responsible for supplying the right information at the right time to the warfighter. This dissertation presents a methodology for automating and realizing time-critical C4ISR applications. We introduce World Wide Web Consortium (W3C) compliant services into the planning and battle management processes where a computer can be more efficient and more effective than a human operator. We demonstrate our approach using ballistic missile defense (BMD) as a case study of a system in which the software services comprising the command, control, and battle management (C2BM) element of the BMD system need to operate within hard real-time constraints. We show the realization of time-critical C4ISR applications via continuously orchestrating individual services based on the automatically processing operational orders (OPORDs) and reports for the system to self-regulate itself. The system monitors, selects, and composes sub-services using a merit-based score until the mission stated in the OPORD is complete. The processing of the OPORDs for use by the C2BM element initiates and preserves the cyclic process of the kill chain used to negate threat ballistic missiles. To select and orchestrate services at runtime, we extended the current Web Services Description Language (WSDL) standard to encompass measures of performance (MOP) and measures of effectiveness (MOE). In our approach the WSDL-advertised measures are continuously updated based on runtime monitoring, creating an historical basis-of-confidence for each of the services. We demonstrate the generation and use by the C2BM of continuously updating service-selection criteria. Our composition language includes a software design pattern for use in ensuring time-critical processes complete within their time budget.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. STATEMENT OF THE PROBLEM....1
B. HYPOTHESIS....2
C. BACKGROUND....3
D. SIGNIFICANCE OF THE PROBLEM....5
E. RESEARCH APPROACH....6
F. CONTRIBUTIONS OF THIS RESEARCH....7
G. OVERVIEW OF THE DISSERTATION....8
H. KEY FINDINGS....9

II. ASSESSMENT OF PREVIOUS WORK....11
A. C4ISR SYSTEMS AND APPLICATIONS....13
B. BMDS AND BMD AGENTS....15
C. MEASUREMENTS....15
D. RDF RDF/XML....16
E. WEB SERVICES STANDARDS AND BPEL....18
1. Web Service Description Language (WSDL)....18
2. SOAP....22
3. Universal Description Discovery and Integration (UDDI)....24
4. Business Process Execution Language (BPEL)....26
F. SERVICE ORIENTED ARCHITECTURE (SOA)....29

III. COMMAND CONTROL AND BATTLE MANAGEMENT....37
A. INTRODUCTION....37
B. APPROACH....38
C. CREATING AN RDF OPORD....38
1. OPORD....38
2. BMDS RDF Vocabulary....41
D. DEVELOPING AND ISSUING RDF OPORDS....49
1. Scenario....50
2. OPORD Header....51
3. OPORD Situation....55
4. OPORD Mission....67
5. OPORD Execution....73
6. OPORD Service Support....78
7. OPORD Command and Signal....80
8. Regional Command Agent Operations Order....82
9. Tactical Command Agent Operations Order....90
10. Summary....93

IV. SERVICE ORIENTED BALLISTIC MISSILE DEFENSE COMMAND CONTROL AND BATTLE MANAGEMENT....95
A. INTRODUCTION....95
B. BMD C2....96

1. C2 Structure Scenario 97
2. Assumptions 98
3. Scenario Execution....98
C. WEB SERVICES FOR BATTLE MANAGEMENT 109
1. Message Types....116
2. Messages....116
D. OPERATIONS ORDER (OPORD)....120
E. BPEL ORCHESTRATION OF BATTLE MANAGER....124
F. EVOLUTION OF OPORDS....129
G. CONCLUSIONS....130
V. PERFORMANCE FEEDBACK MESSAGES....131
A. MESSAGE FORMAT TYPE....132
B. RDF SPOTREP....135
C. CONCLUSION....143
VI. CONCLUSION....145
A. CONTRIBUTIONS....145
1. Proposed Extensions to WSDL Standard to Encompass MOEs and MOPs....145
2. Incorporated MOEs and MOPs to Support Service Selection and Performance Feedback....145
3. Tailored Machine Readable OPORD....145
4. Shadow Pattern, Run-time Monitoring and Performance Feedback....146
B. FUTURE WORK....147
1. Security....147
a. Mult- Level Security(MLS)....147
b. Network....150
c. Coalition....152
2. Real-time Requirements....152
3. Dynamic Selection of Services at Run-Time....153
4. Algorithms for Service Selection....153
5. Statistical Representation of MOEs and MOPs....154
LIST OF REFERENCES....155
INITIAL DISTRIBUTION LIST....159

LIST OF FIGURES

Figure 1. RDF Triple 17
Figure 2. RDF/XML 18
Figure 3. WSDL information model (From: [38]) 22
Figure 4. SOAP message Structure 23
Figure 5. UDDI datatypes (From: [38]) 25
Figure 6. Choreography (From: [39]) 26
Figure 7. Orchestration (From: [39]) 27
Figure 8. BPEL Process as a Web Service (After: [39]) 28
Figure 9. WS Standards Stack (From: [39]) 29
Figure 10. Basic SOA with core Web service standards 35
Figure 11. OPORD Format 40
Figure 12. SCA1 OPORD directed Graph part-1 47
Figure 13. SCA1 OPORD directed Graph part-2 48
Figure 14. SCA1 OPORD directed Graph part-3 49
Figure 15. OPORD Intel Estimate 51
Figure 16. BMDS C2 structure 97
Figure 17. BMDS Scenario 99
Figure 18. Detect 101
Figure 19. Track 103
Figure 20. Assign Weapon 104
Figure 21. Engage 106
Figure 22. Asses Kill 108
Figure 23. Assign Weapon Process 109
Figure 24. Detect Composition WSDL 111
Figure 25. Track Association WSDL 114
Figure 26. Track Correlation WSDL 114
Figure 27. Kill Chain WSDL 115
Figure 28. MOE Roll-Up to MOO 132
Figure 29. SPOTREP (From: [43]) 133
Figure 30. SPOTREP RDF Graph describing Spot Report Tca22sr 140
Figure 31. SPOTREP RDF Graph MOO, MOE, Hit_Ratio 141
Figure 32. SPOTREP RDF Graph Biography Information 142
Figure 33. SPOTREP RDF Graph 143
Figure 34. JCDX Web Services Architecture 149

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Service-Orientation Principles (From: [21]) 32
Table 2. Common characteristics of contemporary SOA (From: [21]) 33
Table 3. Elements and Properties (From: [25]) 43
Table 4. BMDS SCA OPORD 3-tuples 46
Table 5. QoS, MOP, MOE 113
Table 6. Basic Types of Message elements 116
Table 7. Complex types of message elements 116
Table 8. Types of Messages 118

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF LISTINGS

Listing 1. WSDL Structure (From: [14]) 21
Listing 2. SOAP Notification message (From: [17]) 24
Listing 3. OPORD Header 44
Listing 4. BMDS RDF Schema....44
Listing 5. OPORD Header continued....53
Listing 6. Main OPORD paragraphs and annexes 54
Listing 7. OPORD Situation Paragraph 56
Listing 8. Enemy Forces....57
Listing 9. Enemy Forces Most Likely Course Of Action 60
Listing 10. Enemy Forces Most Dangerous Course Of Action....64
Listing 11. Friendly Forces 66
Listing 12. OPORD Mission statement....69
Listing 13. OPORD Priority Defended Asset List and Measures of Effectiveness 71
Listing 14. Measures of Effectiveness 73
Listing 15. OPORD Execution Paragraph....75
Listing 16. Tasks to subordinates....78
Listing 17. Service Support....80
Listing 18. Command and Signal....82
Listing 19. RCA2 OPORD....83
Listing 20. RCA2 OPORD Header 85
Listing 21. Task Organization....86
Listing 22. RCA2 OPORD....88
Listing 23. RCA2 OPORD Execution Paragraph 89
Listing 24. RCA2 Chain of Command....89
Listing 25. TCA 21 OPORD....92
Listing 26. TCA21 Task to Subordinates....93
Listing 27. WSDL AssignWeaponMsg....117
Listing 28. WSDL Application Data....118
Listing 29. WSDL Control Data 119
Listing 30. WSDL Port Type Specs for BM services 119
Listing 31. WSDL Port Type for C2 Third Party Services 120
Listing 32. Operations Order....123
Listing 33. The TCA Process 129
Listing 34. SPOTREP Name Spaces....135
Listing 35. SPOTREP Resource....135
Listing 36. SPOTREP Bio_1....136
Listing 37. SPOTREP Bio_2....137
Listing 38. SPOTREP Internal Resources....138
Listing 39. SPOTREP MOE, MOO, Hit_Ratio 139

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The United States Army afforded me an incredible opportunity to earn a Ph.D. in Software Engineering from the Naval Postgraduate School. While I am grateful and thankful to the Army for the opportunity I could not have succeeded without a tremendous amount of help and support from a number of first-rate professors, students, and friends and the most incredible family anyone could ever have. To these people I am truly indebted and appreciative.

I wish to thank my dissertation committee members Professor Bret Michael, Professor Dan Boger, Dr. Butch Caffall, Professor Doron Drusinsky, Professor Man-Tak Shing, and Professor Duminda Wijesekera for their time, insightful guidance and remarkable patience in teaching and mentoring me in my research. I offer special thanks and gratitude to Professor Michael, Prof Shing, and Professor Drusinsky for all of the extra hours of one-on-one sessions regarding distributed computing and real-time programming. I would like to thank Professor Wijesekera for the many long and late night conversations on Service Oriented Architectures and time-critical Web services. Without these mentoring sessions, I would have been lost. I would like to thank Dr. Butch Caffall for continually encouraging me in this endeavor and for grounding me in software engineering you are a software engineering visionary and one of the finest leaders I have ever met in my 20 years of military service.

I would like to thank Dr. Kevin Greaney for encouraging me to enroll in the software engineering program at the Naval Postgraduate School and for his incredible leadership and mentoring.

I thank Professor Tom Otani, Professor Craig Martell, Professor Richard Riehle, Professor Kevin Squire, and Loren Peitso for all of their wise counsel and support when I really needed it. You are all incredibly gifted educators.

I offer a special thanks to Harsha Tummala an incredibly gifted graduate student from Cal Berkeley and great friend. We spent long hours working together to overcome

many challenges presented in the first prototype of this work. Your contributions to this work are beyond measure. GO DIRTY BEARS!

I thank all of my fellow Ph.D. students for their friendship and help with reviews of my research. I would like thank Commander Kurt Rothenhaus for providing invaluable comments on my research, pushing me to write, and for being a great friend. I would like to thank LTC Mark Orwat and LTC Bill Fischer for their thorough reviews of my dissertation defense presentation it was extremely helpful and big part of my successful defense. You are both great friends. Thanks to CDR Owens Walker and your great family. I wish we had more time to enjoy each other's company and celebrate more NY Yankee victory's.

I thank my Mother and Father who have always been by my side in everything I have ever done. I appreciate and hold dear every phone conversation we have had over the past three years; they helped keep me going. I am who I am because of you and I hope that this achievement makes you proud.

I also must thank my brothers John, Bill, and Steve and their incredible family's for their support and encouragement during my time in school. Your phone calls, family pictures, and emails helped more than you will ever know.

The most important thank you I could ever offer goes to my most precious wife and children. Denise, your support, encouragement, patients, and steadfast love were the reason this work was successfully completed. You have been both mom and dad for our children over much of the past three years and have done an incredible job. I will love you forever with all of my heart. Thank you to my wonderful kids Tommy (Flyin' Russian), Maria (Mia), and Aleksandr (Sosh) you are my inspiration and joy and I love you guys more than you will ever know.

I. INTRODUCTION

A. STATEMENT OF THE PROBLEM

"Today's operator is drowning in information, yet starved for knowledge"[1]

The quote represents how far the Department of Defense (DoD) Command and Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems have come over the past thirty years with respect to collecting and moving information around the battle space. It is also indicative of the need to improve information processing so that the right information gets to the appropriate place at the right moment in time so commanders and systems can make timely decisions and take appropriate actions.

Our research consists of an investigation of how Service Oriented Architecture (SOA) can be used as an enabler to manage the time-critical services which compromise a command and control and battle management parts of a system. SOA provides a number of substantial benefits, including a means for seamlessly integrating software components. Our aim is to reallocate mechanizable and error-prone information processing tasks from the operator to computer systems so the vast amount of information that General Hobbins speaks to can be processed and directed to the apporpriate warfighter for swift action.

Web services (WS) is one *kind* of SOA and the DoD has committed to this implementation, hence applications need to be ported and re-engineered to WS. Because C4ISR applications are such an important component of DoD's business, we focus our research on the design of applications in this domain.

Most WS composition languages are insensitive to Quality of Service (QoS), timeliness, accuracy and a host of other parameters that distinguish C4ISR application from other types of applications.

[1] Gen Tom Hobbins, Air Force deputy chief of staff for war-fighting integration, C4ISR Journal, Aug 1, 2007.

We choose a specific application, Ballistic Missile Defense command and control process and a specific orchestration language Business Process Execution Language (BPEL). We show how to enrich BPEL and WS standards to provide the necessary QoS for the Ballistic Missile Defense (BMD) command structure with a merit-based structure traditionally used to metrisize command structures consisting of

Measures of Performance (MOP)

Measures of Effectiveness (MOE)

Measures of Operational Outcome (MOO)

We introduce design patterns and a performance feedback mechanism to BPEL to ensure time-critical orchestrations complete with in there specified QoS.

B. HYPOTHESIS

Time-critical C4ISR applications can be realized by continuously orchestrating individual services that use automated operations orders (OPORDs) and reports to self-regulate themselves by monitoring, selecting and composing sub-services using a merit-based score until the mission stated in the OPORD is complete.

Network-centric warfare (NCW) is a relatively new theory on warfare that surfaced in 1998 in an article written by Vice Admiral (Ret.) Arthur Cebrowski and John Gartska [1]. The theory has since been further developed by Alberts, Gartska, and Stein in [2] and is now part of The National Defense Strategy of The United States of America under the heading Conducting Network Centric Operations. The theory is based on the following four tenets:

1. A robustly networked force improves information sharing
2. Information sharing enhances the quality of information and situational awareness
3. Shared situational awareness enables collaboration and self-synchronization; and enhances sustainability and speed of command
4. These, in turn, could increase mission effectiveness

C. BACKGROUND

Network-centric warfare is the way the United States will conduct warfare for the foreseeable future. However, the transition or transformation is not going to happen over night and the DoD realizes this. The Office of Force Transformation and Resources (FT&R) under the Office of the Under Secretary of Defense for Policy has been set up to oversee the success of the transformation. The Office of FT&R published [3], which contains a description of NCW and the plan for NCW's implementation. In the conclusion,the report contains three cautions of which the third warns,

> Over time information technology and networking will become commodities. Everyone will have them. At that point, the advantage will go to those best able to exploit those commodities with new organizations and the ability to rapidly change organizations, new doctrine, the ability to create and assimilate technologies with very short cycle times.

The decision to pursue NCW has been made and one of the critical underlying pieces of the infrastructure is the DoD's Global Information Grid (GIG). The GIG is in essence the "network" in the NCW.

It is anticipated that the GIG will realize many of the benefits of NCW. The GIG is one of the DoD's largest and most complex acquisition efforts. The GIG is quintessentially the DoD's own private version of the Internet designed to support Network Centric Operations (NCO). Per [4], the GIG is a "globally interconnected, end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel.

The Defense Information Systems Agency (DISA) is responsible for implementing the GIG. DISA, in concert with the DoD Chief Information Officer (CIO), chose a Service Oriented Architecture (SOA) approach and Web services (WS) for implementing DoD information systems, as outlined in [4].

In addition to the GIG, DISA is responsible for leading the development of the next generation C2 systems called the Net-enabled Command Capability (NECC),

formerly the Joint Command and Control (JC2). The NECC too is being developed with the WS and BPEL technology. The effort

> would not look to its past for its inspiration, past of large monolithic systems colloquially known as stove-pipes, but look forward to the future where capability was distributed across a powerful, flexible, and redundant network. It would be based on small modular components (i.e. services) organized into capability modules that are meant to be agile and adaptable.[5]

The description of NECC and the description of the GIG are at the heart of a SOA: services on a network that are discoverable, composable, loosely coupled, autonomous, based on standards, extensible, and more described later. In such an environment a capability like NECC can offer more utility to more people than stove-piped systems in which you get more of a "what you see, is what you get". Still though, the underlying technology used to implement the SOA must support the DoD's needs. As mentioned above DISA has chosen WS as one of its implemention technologies and while we believe a WS implementation of a SOA is as good as any, it is known that WS lacks certain capabilities such as those described in [7].

According to [7], one of the weaknesses of the current implementations of WS is the inability to account for time. Time, in most military operations, is critical. Time and temporal aspects are fundamental in the development of all military operations orders and plans. The U.S. Army acronym METT-T stands for the factors to be considered in planning any tactical mission: Mission, Enemy, Troops, Terrain and weather, and Time, available. None of these factors, especially time, is optional. There are several examples of bad timing in the warfighting domain: receiving a Chemical, Biological, Radiological, and Nuclear (CBRN) report seconds after entering a contaminated area, receiving enemy locations after it is too late to avoid an ambush, and laying fire on vacted locations because intelligence reports about enemy troop movements were received too late.

The point is that time and temporal aspects are critical in military operations. In this dissertation we investigate the ability to reason about, specify, and enforce timing and temporal properties in DoD WS applications.

D. SIGNIFICANCE OF THE PROBLEM

An enabling technology that does not support timing and temporal aspects could disqualify both the entire systems and services in addition to the data and information they contain or produce from residing on the GIG or isolate them in a manner such that they are, for all intent and purposes, stovepipes. This is not the intent of either NCW or the GIG. The time-sensitive systems and services we refer to could include any number of existing C4ISR systems to include the NECC mentioned above; each can produce and share volumes of time-sensitive data and information, given the chance.

Let's consider an example of a time-sensitive system of systems (SOS): the Ballistic Missile Defense Systems (BMDS) described in detail in [9]. The BMDS consists of legacy systems (e.g., the U.S. Navy's AEGIS weapon system and U.S. Army's THAAD system) and new developments (e.g., the Ground-Based Midcourse Defense (GMD) system's organic Sea Based X-Band Radar (SBX)) connected together and managed by the Global Integrated Fire Control (GIFC, but now know as the Global Engagement Manager (GEM)) system. This SoS will serve as a global shield against threat ballistic missiles. The BMDS functions based on a kill chain for which the system's duty cycles have hard real-time requirements; the timing requirements are driven to a large extent by the travel time of a threat missile from launch to its targeted destination. The kill chain consists of five concurrent activities: (i) Detect, (ii)Track, (iii) Assign, (iv) Engage, and (v) Assess. The BMDS detects the launch of an object. Detected objects are then tracked with the aim of determining the nature of the object: is it benign or a threat. The system assigns weapon systems to engage threat objects, manages the engagement, and then assesses the success of the engagement and based on that assessment the next action to take (e.g., if the weapon missed the threat object then determine whether to assign another weapon to engage the threat object, such as with a shoot-look-shoot doctrine). Much of the decision-making that goes on within the BMDS happens in the Battle Manager portion of the GIFC. To make our point about the importance of time and temporal aspects we look at track processing, a function within the track portion of the kill-chain. For our example we treat track processing as a process that uses three primary services: a discriminator, a correlator, and a local clock. The

"track process" process receives tracks from the senors, and then calls a discriminator service. Based on the results reported by the discriminator the process either assigns a "no-kill" (i.e., the object is benign) or assigns a kill (i.e., the object is a threat) and then calls a correlator service. The correlator checks whether the object is attributable to any existing tracks. The entire "track process" process has a complete-not-later-than time associated with it as the process must finish in support of the whole kill chain cycle time.

The process needs to determine how long it will take each of the supporting services (i.e., discrimination and correlation) to complete their their tasks: If these services advertise their expected execution times then the track process application can use this information to select from the available pool of discrimination and correlation services that best meet the system's timing requirements. Once the "track process" process selects its services the process must be monitored and call exceptions when a process exceeds the specified deadlines. Moreover, the selection of appropriate services also depends on the quality of the solution provided by the services. The discriminator and correlator services need to provide a measure of their effectiveness in carrying out their tasks. Currently SOA does not provide the constructs needed for the process such as the track process to select services based on the timing and effectiveness properties of the services.

E. RESEARCH APPROACH

In our assessment of previous work we found that some of the chosen development technologies, namely WS and BPEL, for DoD programs such as the GIG and NECC lack the constructs and supporting infrastructure necessary to specify, check, and enforce timing properties. As explained above, in the significance of the problem section, timing properties, QoS, MOP, and MOE are critical in many C4ISR systems throughout DoD and must be correctly specified and implemented in their WS application.

In our research, we extend WSDL and BPEL syntax to include timing deadlines, QoS, MoP, and MOE for the basic services, their choreography and alternative exception handlers that ought to be invoked when timing faults occur. We introduce a machine

processable Operations Order (OPORD) tailored to support the MOP and MOE for assigned missions. We then introduce a design pattern into the BPEL to ensure that all orchestrations complete in the necessary time and according to associated QoS, MOPs, and MOEs. Finally, we introduce runtime monitoring and performance feedback mechanisms for BPEL orchestrations and services. This provides a capability to ensure continued execution when a process exceeds the specified deadlines and to provide updated QoS, MOP, and MOP information back to the developers and consumers for updating advertised WSDL data and analysis. We show the utility of our extensions and their runtime by designing a prototype of the GIFC-like battle manager.

F. CONTRIBUTIONS OF THIS RESEARCH

As described above WS, BPEL and the supporting runtime infrastructure lack timing constructs, QoS, MOP, and MOE necessary to specify, reason about, and enforce timing and temporal properties to meet the needs of DoD WS applications. We propose a methodology for modeling specific time-critical C4ISR applications that includes the following contributions:

1. **Propose extensions to WSDL standard:** We propose extending current WSDL standards to encompass MOEs and MOPs. These extensions allow services to advertise what functions they perform and how well they perform the functions based on QoS, MOPs, and MOEs.

2. **Incorporated MOO, MOE and MOP:** Incorporating these measures supports the selection of complex services at runtime and facilitates performance feedback upon completion of the services execution.

3. **Tailored machine processable OPORD for time-critical orchestrations:** The introduction of a machine processable OPORD initiates the entire execution of the application. Humans write OPORDs but the OPORD can be processed entirely by machines (battle managers in our case study).

4. **Introduction of the shadow design pattern, runtime monitoring, and performance feedback:** To address time-criticality in BPEL we introduce the

shadow design pattern similar to the approach used in [32]. In addition, we introduce a performance feedback service to build a basis of confidence in available services by maintaining a running record of each service's QoS, MOPs, and MOEs. These histories can then be used by clients to aid in selecting the appropriate service for their particular task or application

As a proof of concept we prototyped a portion of a missile defense system's battle manager along with a set of stubbed services that the battle manager can utilize.

G. OVERVIEW OF THE DISSERTATION

In Chapter II, we provide background infrormation on six topic areas essential to understanding this research: C4ISR systems and applications, BMD and BMD Agents, SOA, Resource Description Framework (RDF), and measures, metrics, and QoS. We provide a brief history of the evolution of DoD C4ISR systems and applications. We show the significance of the role of C4ISR systems and applications in Network Centric Warfare (NCW)- way the United States will organize and fight in the information age [2]. We then highlight some of the challenges in the area of SOA based that we beilive our research can help to evolve and advance the state of the C4ISR systems and applications to better support NCW.

Chapter III details machine- and human-readable Operation Order (OPORD) prototypes. Creating machine "understandable" OPORDS and reports is essential in our research as we try to keep humans from having to process large volumes of data correctly in time-constrained situations.

We briefly describe a typical five-paragraph operations order as defined in [24]. The OPORD is the document that all units and systems use for initialization and synchronization prior to expected hostilities. The order states the mission and commanders intent and provides detailed instructions to to subordinate units.

Next, we show three operations orders written in RDF/XML. In this machine-readable form, the orders or pieces of the orders can be exchanged in a message format and processed in a fully automated manner.

Chapter IV shows a prototype of our merit-based architecture for the automatic selection and composition of services in a SOA-based C4ISR system (the BMD SOA for our case study). We describe the BMD Command and Control (C2) hierarchy, define a couple prototype WS for battle management, and show the orchestration of a kill-chain process, a duty cycle for negating enemy ballistic missiles.

Chapter V describes our prototype for the performance feedback service used to build the historical basis of confidence for all participating orchestrations and services.

Chapter VI highlights the contributions toward software engineering and future directions of our research.

H. KEY FINDINGS

In this research, we show how the BPEL with appropriate extensions for MOOs, MOEs, MOPs and QoS parameters can be used to specify C2 and battle management needs of BMD.

Specifically, we incorporated MOOs, MOEs, and MOPs to support the selection of complex services at runtime and to provide feedback on a services performance upon completion of execution. This contribution formally ties metrics to the executable software and provides support for guiding dynamic composition.

The tailored machine processable operations order supports time-critical orchestrations. This contributions transforms a once slow, highly manual, and error prone planning process into a much faster process with reduced ambiguity and increased precision.

Finally, the introduction of the shadow pattern, runtime monitoring, and performance feedback address the lack of support for time constrained orchestrations in the BPEL.

THIS PAGE INTENTIONALLY LEFT BLANK

II. ASSESSMENT OF PREVIOUS WORK

In this chapter we give a set of definitions that are used for our research, provide some background information on essential concepts and technology used in our research, and provide a brief assessment on some previous work.

Definitions: The following terms: System-of-Systems (SoS), Global Information Grid (GIG), Network Enabled Command and Control (NECC), and Ballistic Missile Defense System (BMDS), are briefly discussed below before we describe our assessment of previous work to help put our research into perspective. We will provide greater detail on the BMDS as it is used as our case study for the dissertation.

System of Systems (SoS): The definition of a SoS is captured best in [9] where Caffall points out that SoS presents unique challenges to developers not seen before in traditional systems engineering. Caffall defines a SoS as "an amalgamation of legacy systems and developing systems that provide an enhanced military capability greater than that of any of the individual systems within the system-of-systems." We adopt this definition for use in our work as it sufficiently captures the essence of NCW and the association between SoS and NCW.

Global Information Grid (GIG): Per [4], The GIG is "The globally interconnected, end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating and managing information on demand to warfighters, policy makers, and support personnel." The GIG includes all owned and leased communications and computing systems and services, software (including applications), data, security services, and other associated services necessary to achieve Information Superiority. It also includes National Security Systems as defined in section 5142 of the Clinger-Cohen Act of 1996 (reference (b)). The GIG supports all Department of Defense, National Security, and related Intelligence Community missions and functions (strategic, operational, tactical, and business), in war and in peace. The GIG provides capabilities from all operating locations (bases, posts, camps, stations,

facilities, mobile platforms, and deployed sites). The GIG provides interfaces to coalition, allied, and non-DoD users and systems.

The GIG is the network and other infrastructure that facilitates communication, information sharing, and understanding between all entities across the DoD related communities of interest. To ensure that the multitude of systems, applications, and services participating in the GIG are interoperable DISA decided to implement GIG Network Centric Enterprise Services (NCES)[2] using a SOA.

Network Enabled Command and Control (NECC): NECC is the DoD's next generation joint command and control (C2) system. The significance of the NECC program for this research rests in the fact that NECC is the first new C2 program that is being developed using a "net-centric architecture" using WS and BPEL as its SOA implementing technology. As stated earlier neither the WS nor BPEL, in their current form, has the appropriate construct to specify or check needed timing requirements.

Ballistic Missile Defense System (BMDS): The BMDS is a DoD program under the direction of the Missile Defense Agency (MDA). The Missile Defense Agency's mission:

> …is to develop an integrated, layered Ballistic Missile Defense System to defend the United States, its deployed forces, allies and friends from ballistic missiles of all ranges and in all phases of flight.[11]

The BMDS is a SoS made up of component systems connected by a network. The component systems range in variety of types of legacy and developing sensors, weapon systems, and GIFC battle management systems. The component systems are geographically dispersed across the globe.

The component systems are logically associated with a single GIFC on the BMDS Network from which each component receives battle management tasks (e.g. a GIFC may task a weapon system to fire on a particular target while another weapon system in the logical grouping may be tasked to hold fire on that same target, a sensor in the logical

[2] NCES are web services built on a SOA foundation, the "SOA Foundation provides reliable and interoperable capabilities that will enable services-based applications to take advantage of existing network components." [10]

group may be tasked to radiated energy on some other object to help identify it). The GIFC of each logical group of components is responsible for efficiently managing its set of component sensors and weapons systems based on an operations order received from a "leader" GIFC.

The GIFCs manage their components based on the operations order, communications with the individual components, and communications with the "leader" GIFC. Typical information passed between a GIFC and its components are health and status information (e.g. weapons system is dead lined (not usable), weapon system is in service, number and types of missiles remaining, sensors current orientation, weapons system in engagement, etc) and battle management tasks such as the ones described in the preceding paragraph. The GIFCs use this type of information to efficiently manage their components (i.e. make sound decisions on what sensors should look where and when, what weapons systems should fire, what weapons systems should be on stand-by, what weapons systems should be in a hold-fire status).

Another source of input for battle management comes from the "lead" GIFC. Initially, all of the GIFCs are assigned a mission of a single integrated operations order. Each GIFC is responsible for executing its mission within the order. The "leader" GIFC monitors execution of the operations order and manages all components from a global perspective, through the components associated GIFC. The "lead" GIFC receives information from the subordinate GIFCs during execution of a battle and assesses it against the operations order, if the "leader" GIFC determines the plan is no longer tenable, it issues commands to the subordinate GIFC's to bring the battle back into a favorable state (i.e. all targets are destroyed).

A GIFC-like battle manager prototype, as described in the problem statement, is ideal for use as a case study in our research as it is representative of many command and control systems throughout the DoD.

A. C4ISR SYSTEMS AND APPLICATIONS

Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) systems and applications have always played a significant role

in the Department of Defense (DoD). C4ISR applications and systems produce and consume vast amounts of information for the purpose of enhancing the ability and capabilities of decision makers up and down the chain of command.

Since the origins of warfare the methods and technologies for collecting, processing, and disseminating has changed and for the most part improved dramatically. Less than one hundred fifty years ago many orders, commands, and reports were written or given verbally to messengers on horseback to deliver between units; this method was much improved over the previous hundred years, but was not without challenges. One such challenge was the ambiguity of the written word or the loss of meaning in the translation from the messenger and the time it would take to get clarification. This is best illustrated in [36] as it describes the ambiguities in the messages exchanged between General Lee and General Stewart in the events leading up to and including the Battle of Gettysburg.

The military continuously tries to find new ways to reduce both the time to communicate orders and reports. It also strives to improve the format of the OPORD to convey as unambiguously as possible the commander's mission and intent and the tasks units are to execute. The time to communicate the orders and reports has also seen significant improvements. The means of communication has gone from horseback to telecommunications to high-speed satalite networks. The advent of high-speed networks and the number and variety of systems that produce information can present large volumes of information to operators and analysts who then must filter that information so that commanders can make sound and informed decision, but as [35] points out the amount of information is overwhelming. We show in our research a design to reduce the information available into relevant and useable information in a timely manner. In addition, we reduce ambiguties to improve command and control, and battle management. Our approach requires the automation of certain tasks, some processing tasks and decision-making tasks that are currently conducted by soldiers.

B. BMDS AND BMD AGENTS

In [29] Wijesekera et al. present an agent-based framework to model BMDS C2 strategies with the BMDS modeled as a "distributed system of interacting agents." The framework treats doctrine, policy and organizational structure as rule-based constraints on system behavior to support both analytical and simulation modeling of alternative C2 strategies based on scenarios.

The centerpiece of this research is the methodology presented to "systematically model the organizational structures that assess the Measures of Effectiveness (MoE) and Measures of Performances due to cause-and-effect relationships inherent in distributed decision making [29]." The organizational structures are modeled as a distributed "team" of decision makers that adhere to strict doctrine and policy. The models are then exercised to compare different C2 structures for use in BMD, such as those identified in [37].

We extend this research by using the methodology and implementing the proposed model as services and service orchestrations in a SOA based BMDS.

C. MEASUREMENTS

Measurements and metrics play a significant role in this research. They are used to aid consumers in selecting the most suitable services for their applications based on a service's advertised QoS, MOP or how a service performed against previous consumers' MOE. We propose a design to support a basis of confidence for services and orchestrations of services (a service itself). The basis of confidence is a history of how the services performed both in their local environment, i.e., on the services application server and in context of the application mission, i.e., a services result used in an application. The basis of confidence is contiuously updated as the service is used.

The basis of confidence profile grows as the service is used. Initially, service consumers could select services based on MOPs. MOPs are results from developer's intial testing, benchmarks, etc. and later on, as its use increase, its actual performance. A services MOPs are performance indicators with respect to the service running on its server. Examples of advertisable attributes include a service's availability, reliability,

execution time, and precision of result; these are made available to give consumers an idea of expected, but not guaranteed, performance as there is no context information nor does it include information regarding network performance. Each time a service is invoked monitors record how the service performed in context; this data can be added to the basis of confidence giving potential users of a service information on how a service performed in context.

The definitions of MOE, MOP, and QoS for this research follow.

> Measure of Effectiveness (MOE) — A criterion used to assess changes in system behavior, capability, or operational environment that is tied to measuring the attainment of an end state, achievement of an objective, or creation of an effect [30].
>
> Measure of Performance (MOP) — A criterion used to assess friendly actions that is tied to measuring task accomplishment [30].
>
> Qaulity of Service (QoS) — The non-functional characteristics of a service or orchestration of services sought or offered by a consumer or developer. Availability, reliability, operational latency, and security are but a few examples of QoS that a service may or may not exhibit. We use the work in [31] as the basis for our definition.

D. RDF RDF/XML

RDF is a set of eXtensible Markup Language (XML) based specifications that describe a data model that is grounded in graph theory and are used to describe resources on a network. The fundamental construct of the data model is an RDF triple; three pieces of information that define a single bit of knowledge: the subject, predicate, and object. It is no mistake that these same three pieces information describe an English sentence or statement. The triple is the enabling mechanism that permits "human understanding and meaning to be interpreted consistently and mechanically"[25] and is at the heart of the data model.

Figure 1 below is the graphical representation of an RDF triple that represent the English statement; http://swe.nps.edu/bmds/OPORDS/SCA1opord20080129 has a classification level whose value is "Unclassified'. The graph shows the subject represented by the URIref http://swe.nps.edu/bmds/OPORDS/SCA1opord20080129 in

the oval, the predicate represented by the URIref http://swe.nps.edu/bmds/elements/1.0/classification, and the object represented by the literal value “Unclassified” in the rectangle.

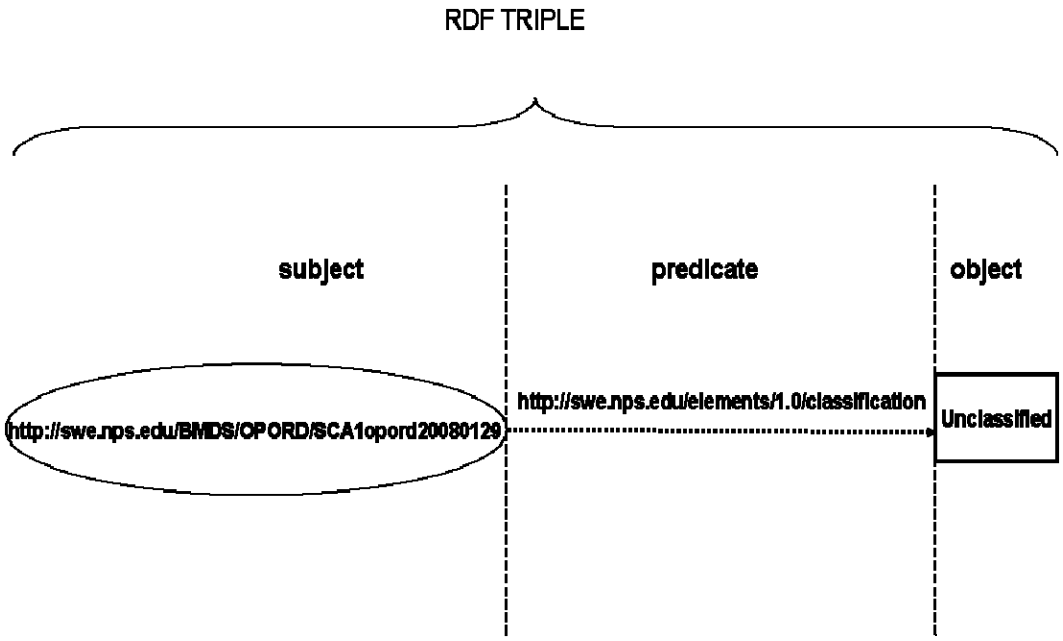

Figure 1. RDF Triple

RDF represents triples as labeled directed graphs. The nodes represented in the graphs can be one of three types: a Uniform Resource Identifier (URI) reference (URIref) indicated by a rectangle or an oval, such as the subject shown in the graph; a literal encompassed in a rectangle node as represented by the object above in the graph; and a blank node not shown in this particular graph, but would be represented by a blank circle. The connecting arc is labeled with a predicate and is always incident from the subject node and incident to the object node. Both nodes and arcs can be labeled with a URIref as described above. URIrefs play a significant role in RDF graphs as they are the means used to identifying resources uniquely. A URIref can function as both a name and location as we will see in the development of our OPORD prototype. The RDF directed graphs describe the RDF data model and are very convenient and easy to read for

humans, but they are not particularly useful for computers in this form. For this task, the RDF specification describes its official serialization technique as RDF/XML.

The RDF/XML is the XML syntax for RDF that encodes RDF graphs. Figure 2 below is the RDF/XML of the graph in above. The subject, http://swe.nps.edu/bmds/opords/SCA1opord20080129, is shown at line four. The predicate http://swe.nps.edu/bmds/elements/1.0/classification, is captured in the classification tag at line five. Finally, the object literal, "Unclassified", is also located in line five between the classification tags.

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/199/02/22-rdf-syntax-ns#
3   xmlns:bmds="http://swe.nps.edu/elements/1.0/">
4 <rdf:Description rdf:about="http//swe.nps.edu/bmds/opord/opordsca20080129">
5    <bmds:classification>Unclassified</bmds:classification>
6 </rdf:RDF>
```

Figure 2. RDF/XML

E. WEB SERVICES STANDARDS AND BPEL

In this research we investigate the WS type of SOA and therfore briefly describe its core set of standards commonly refered to as the WS stack (WSDL, SOAP, and UDDI). We also describe BPEL because BPEL is the standard we have chosen for the coreography of multiple services. Each standard plays a significant role in a SOA. After describing the individual standards we provide a simple illustration to show how they work together to create the rudiments of a SOA.

1. Web Service Description Language (WSDL)

WSDL is a XML-based language used to describe WS. The WSDL serves as contract between the WS and a consumer or potential consumer of that service. The WSDL file describes both the data to be passed and the method for passing the data.

The first section of a WSDL defines the information that is to be sent to and from a service using XML Schemas and the <types> tag. Next, the types created previously are used to define the content of the messages that are to be exchanged between the service and its consumer and use the <messages> tag. The next part of the WSDL is identified by the <portTypes> tag. <portTypes> is akin to an interface as it defines the operations that the service provides. The operations can be either input or output operations and consist of the messages we discussed above. Next, the WSDL defines the protocol that is to be used to send messages. The protocol information is captured in the <binding> element tag. Within the <binding> tag can be a number of attributes that identify how communications are carried out. For instance, the following tag <soap:binding style =”rpc” transport=”http://schemas.xmlsoap.org/soap/http” /> indicates that the message is a remote procedure call (rpc) type SOAP message and the message is being sent over HTTP. The style could also have a value of “document,” allowing for the exchange of document type messages that are typically more complex than rpc types. Inside the binding tag is the <operation> tag which is a container for the <soap:operation> that simply states that the message is a SOAP message. Within the <soap:operation> are the <input>, <output> or in many cases both tags which contain the <soap:body> tag which can contain a several attributes describing the message. The example

<soap:body use=”encoded” encodingStyle=”http://www.w3.org/rdf-encoding” namespace=”http://www.swe.nps.edu/opord ” />

indicates that the input message is a soap message and the use value “encoded” means that the server assumes the RDF/XML meaning as identified by the encodingStyle value. The last piece, namespace, identifies the namespace of the body of the SOAP message. The <service> element tag completes the major portions of the WSDL. The <service> element specifies where and how to send information. An example of the service definition follows:

<service name=”correlationService”>

<port name=”correlationPort” binding=”namespace:correlatorBinding”>

```
                <soap:address location=http://swe.nps.edu/correlator” />
            </port>
        </service>
```

The port element above is the endpoint for the connection between the server and a consumer. The binding is identified and then sent as a SOAP message to the address identified by the location attribute.

Listing 1 below is the grammar of the WSDL structure taken from [14] .

```
1: <wsdl:definitions name="nmtoken"? targetNamespace="uri"?>
2:     <import namespace="uri" location="uri"/>*
3:     <wsdl:documentation .... /> ?
4:
5:     <wsdl:types> ?
6:         <wsdl:documentation .... />?
7:         <xsd:schema .... />*
8:         <-- extensibility element --> *
9:     </wsdl:types>
10:
11:     <wsdl:message name="nmtoken"> *
12:         <wsdl:documentation .... />?
13:         <part name="nmtoken" element="qname"? type="qname"?/> *
14:     </wsdl:message>
15:
16:     <wsdl:portType name="nmtoken">*
17:         <wsdl:documentation .... />?
18:         <wsdl:operation name="nmtoken">*
19:             <wsdl:documentation .... /> ?
20:             <wsdl:input name="nmtoken"? message="qname">?
21:                 <wsdl:documentation .... /> ?
22:             </wsdl:input>
23:             <wsdl:output name="nmtoken"? message="qname">?
24:                 <wsdl:documentation .... /> ?
25:             </wsdl:output>
26:             <wsdl:fault name="nmtoken" message="qname"> *
27:                 <wsdl:documentation .... /> ?
28:             </wsdl:fault>
29:         </wsdl:operation>
30:     </wsdl:portType>
31:
```

```
32:    <wsdl:binding name="nmtoken" type="qname">*
33:       <wsdl:documentation .... />?
34:       <-- extensibility element --> *
35:       <wsdl:operation name="nmtoken">*
36:          <wsdl:documentation .... /> ?
37:          <-- extensibility element --> *
38:          <wsdl:input> ?
39:             <wsdl:documentation .... /> ?
40:             <-- extensibility element -->
41:          </wsdl:input>
42:          <wsdl:output> ?
43:             <wsdl:documentation .... /> ?
44:             <-- extensibility element --> *
45:          </wsdl:output>
46:          <wsdl:fault name="nmtoken"> *
47:             <wsdl:documentation .... /> ?
48:             <-- extensibility element --> *
49:          </wsdl:fault>
50:       </wsdl:operation>
51:    </wsdl:binding>
52:
53:    <wsdl:service name="nmtoken"> *
54:       <wsdl:documentation .... />?
55:       <wsdl:port name="nmtoken" binding="qname"> *
56:          <wsdl:documentation .... /> ?
57:          <-- extensibility element -->
58:       </wsdl:port>
59:       <-- extensibility element -->
60:    </wsdl:service>
61:
62:    <-- extensibility element --> *
63: </wsdl:definitions>
```

Listing 1. WSDL Structure (From: [14])

We wrap up our description with a figure of the WSDL information model taken from [38]. Figure 3 captures the essence of the separation between the abstract specification and the concrete implementation that supports the reuse of the abstract definition of the service.

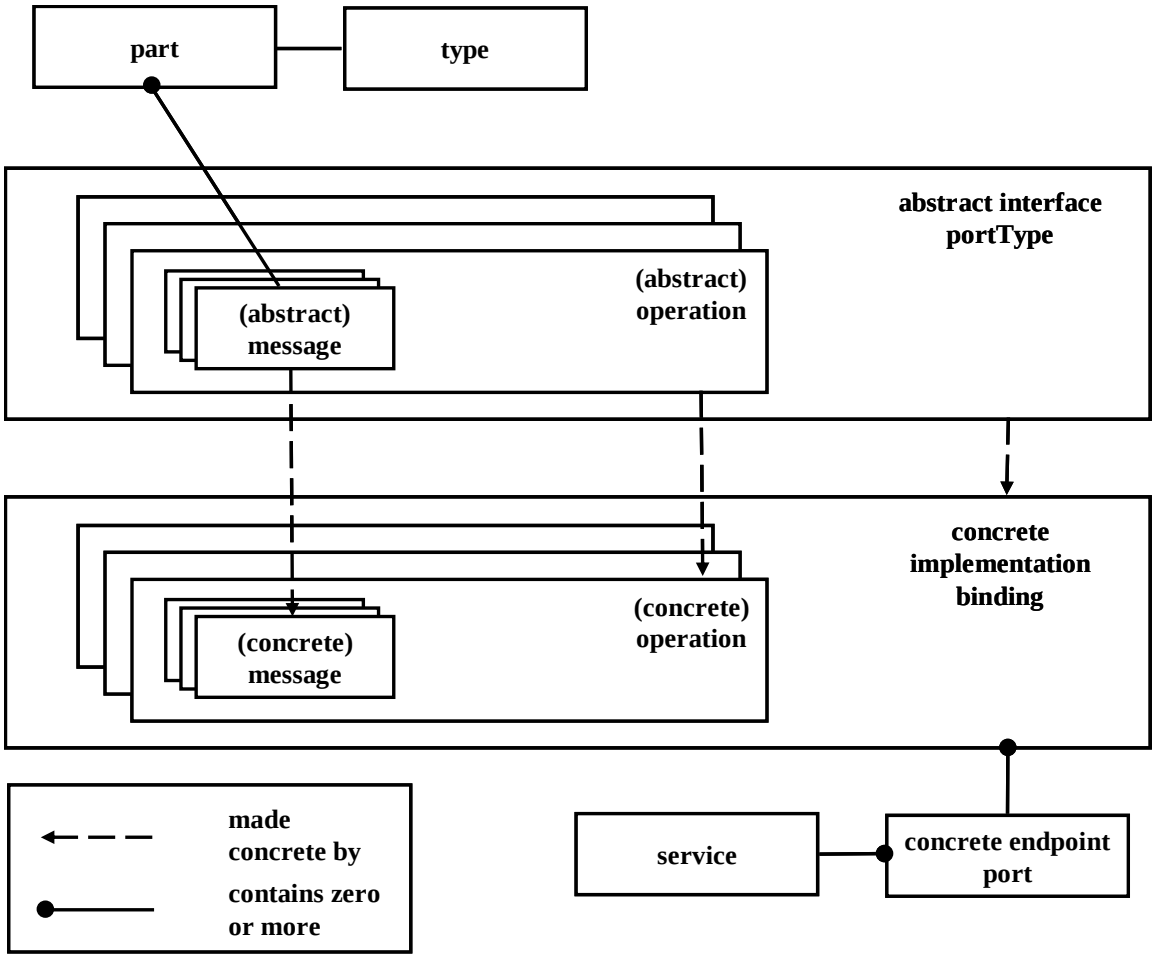

Figure 3. WSDL information model (From: [38])

2. SOAP

SOAP is an XML-based language that serves as a standard wrapping protocol for messaging used in communicating across a network using any network protocol, but the most popoular network protocol typicaly used is HTTP. When data is sent to a SOAP server it must be specified in a particular manner. SOAP messages have three basic parts the SOAP <envelope> element, which contains a SOAP <Header> element and and a SOAP <body> element. The SOAP <Header> is an optional element of the envelope, but if it exists there can only be one and it must be the first child element of the envelope. The SOAP <body> element can contain any XML that is well formed and namespace-qualified, but it cannot have any Document Type Definition (DTD) or processing instructions. The SOAP <header> element can contain SOAP <header block> elements.

The header blocks contain contextual information for processing the SOAP message. The SOAP message is quite simple, as seen in Figure 4.

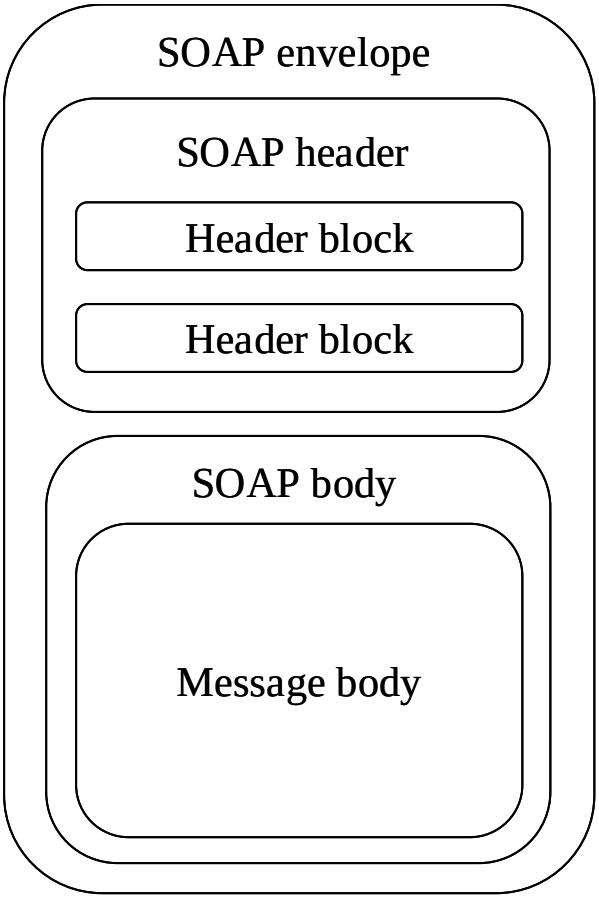

Figure 4. SOAP message Structure

We provide a sample notification message expressed in SOAP taken from the SOAP 1.2 standard [17]. The message in Listing 2 contains the elements shown above in Figure 4. The SOAP Envelope starts at line 1 and concludes with its end tag at line 13. Within the SOAP Envelope we see the first child element; the SOAP Header starting at line 2 and ending at line 7. Inside the SOAP Header there is a header block that starts at line 3 and ends at line 5. The header block in this example contains application-specific data. Line 3 contains an alertcontrol identified at the namespace given. Lines 4 and 5 contain a priority and epiration date-time group. As [17] points out,

> In general, SOAP header blocks contain information which might be of use to SOAP intermediaries as well as the ultimate destination of the

message. In this example an intermediary might prioritize the delivery of the message based on the priority and expiration information in the SOAP header block.

The next element in the SOAP Envelope is the SOAP body, lines 8 through 12. The SOAP body contains the actual message payload, that is, the message body, which in this case is the alert message at line10.

```
1: <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
2:    <env:Header>
3:       <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
4:          <n:priority>1</n:priority>
5:          <n:expires>2001-06-22T14:00:00-05:00</n:expires>
6:       </n:alertcontrol>
7:   </env:Header>
8:   <env:Body>
9:      <m:alert xmlns:m="http://example.org/alert">
10:       <m:msg>Pick up Mary at school at 2pm</m:msg>
11:    </m:alert>
12:   </env:Body>
13: </env:Envelope>
```

Listing 2. SOAP Notification message (From: [17])

We have given a basic treatment to what SOAP is and what the message framework looks like. For a detailed description of SOAP interested readers should see [17], [18], and [19].

3. Universal Description Discovery and Integration (UDDI)

UDDI is also an XML-based language. It is provides the necessary information to permit WS to be registred in a database registry so that they might be discovered by potential service consumers. UDDI is a registry of descriptions of services available for use much like telephone yellow pages provides information about available services. The registry is a hierarchical structure of business, service, and binding information represented in XML. The purpose of UDDI is to make service discovery possible at design time and dynamically at runtime.

There are two main types of UDDI registries. The first is a public registry called the UDDI Business Registry (UBR). The UBR is a set of UDDI nodes hosted by individual organizations across a network, with replication of the registries. There is also a private registry. The private registries are typically hosted on intranets for use within an organization.

The information in a UDDI registry consists of five basic datatypes businessEntity, businessService, bindingTemplate, tModel, and publisherAssertion. We borrow the figure below from [38] to explain the relationship of the types listed above.

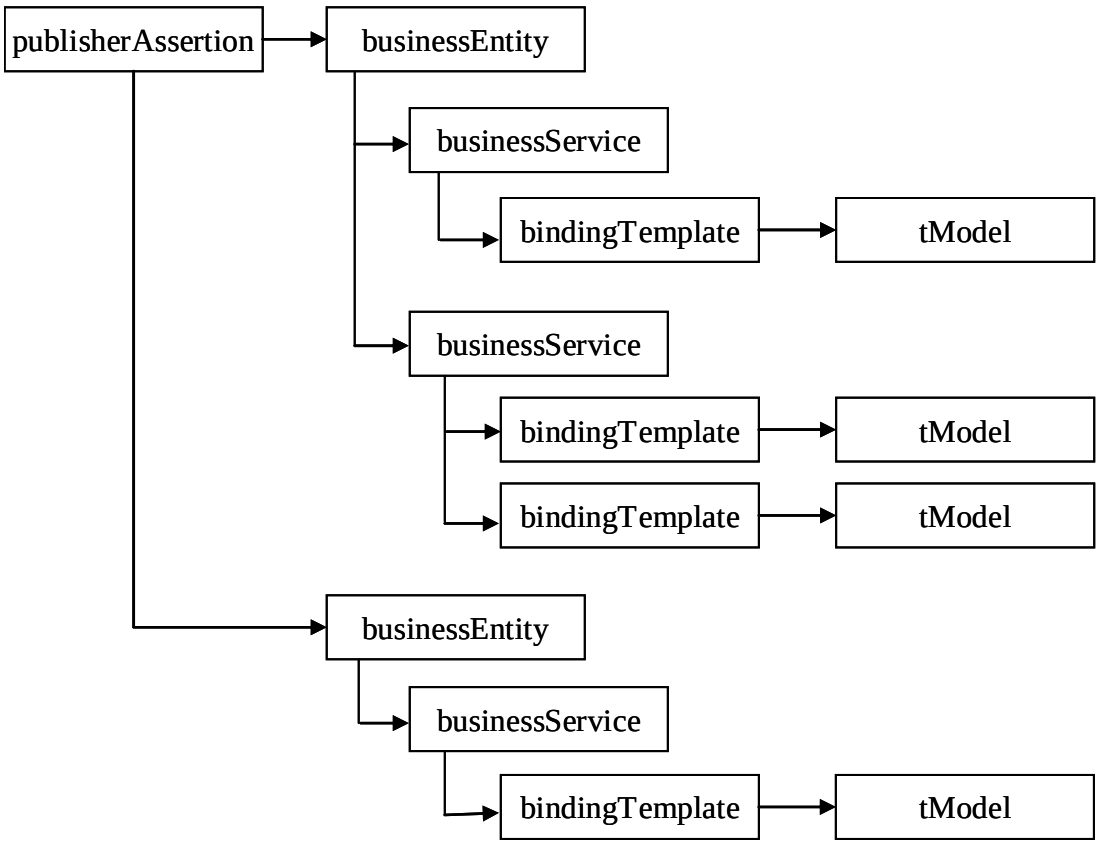

Figure 5. UDDI datatypes (From: [38])

The datatypes shown above in Figure 5 are related in the following manner. The businessEntity element identifies information about a business and can have references to one or more businessServices. The businessServices and its bindingTemplate elements

contain the technical and business descriptions for a WS. Each bindingTemplate element contains a reference to one or more tModels; these references describe the technical specifications of a service. Finally, the publisherAssertion element defines the relationship between two businessEntity elements.

The tModel datatype plays a significant role in UDDI. The tModel references the WSDL document that specifies the technical details needed to invoke the service.

4. Business Process Execution Language (BPEL)

Like the WS Core standards, BPEL is an XML-based standard. It is a standard used to compose WS and automate business processes. Composition occurs in BPEL in two ways; coreography or orchestration. We briefly describe the differences and then delve further into orchestration as it tends to be a more flexible means of composition and suits our research better.

Choreography is the composition of web services in a manner where each WS knows exactly when it executes its operations and who or what it is interacting with as shown below in Figure 6.

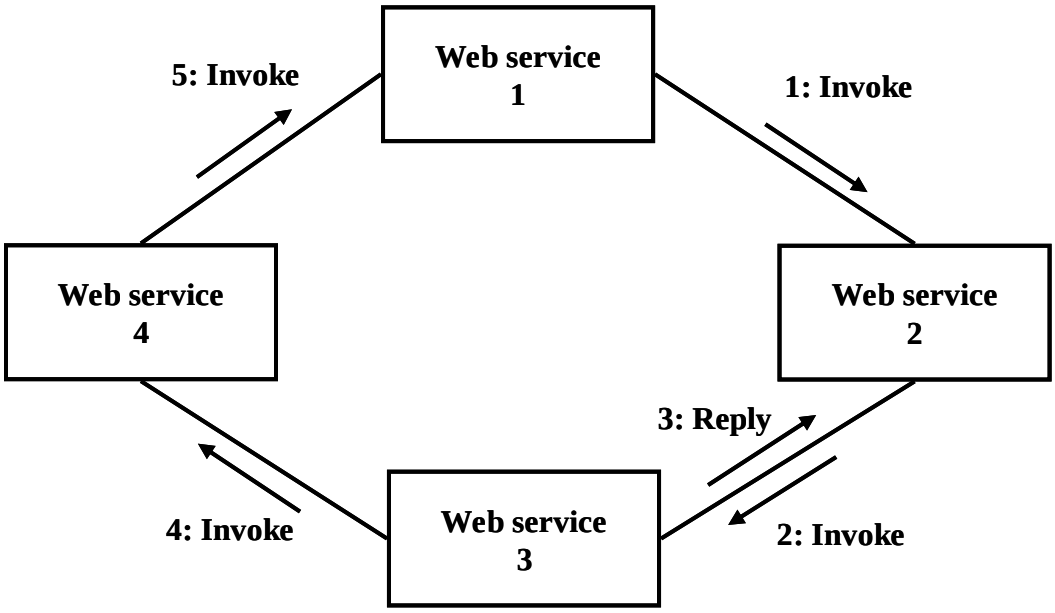

Figure 6. Choreography (From: [39])

When composing WS using orchestration a central process controls and coordinates what operations of what WS are executed and when. The WS involved do not know that they are participating in a higher level process. The orchestration process is show below in Figure 7.

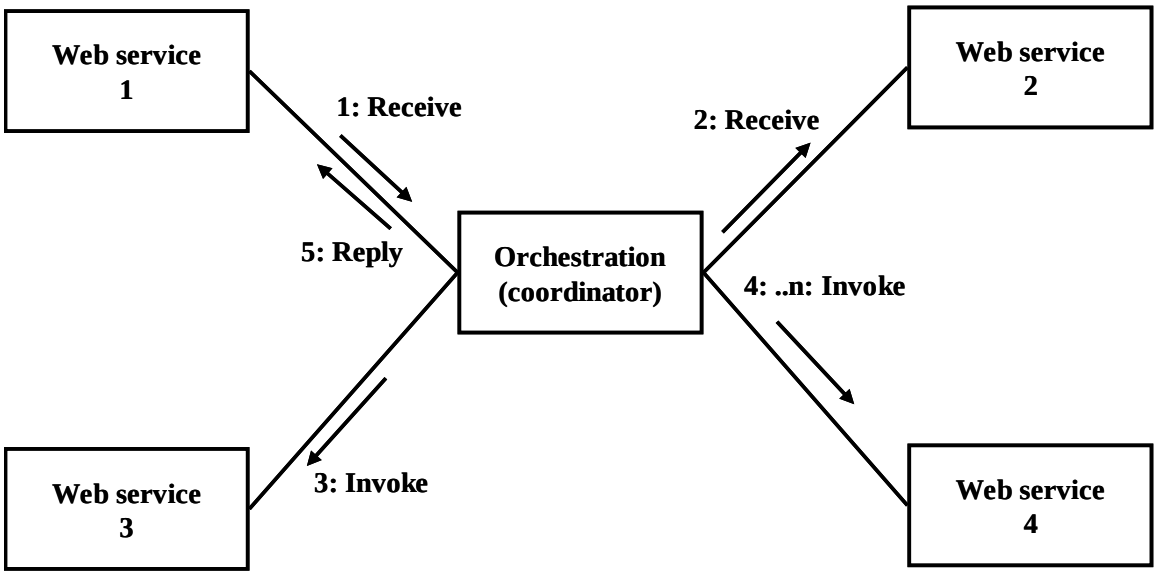

Figure 7. Orchestration (From: [39])

BPEL is capable of specifying both types of WS composition. For the choreography approach, known as the abstract business process in BPEL, the message exchange between WS is specified and the details of the process flows are omitted; this approach does not allow for the execution of the process. In the BPEL orchestration approach or Executable business process, the details of the entire business process are specified and the business process can be executed by a BPEL execution engine.

Compared to the Abstract approach, the BPEL orchestration approach offers more flexibility as it allows the selection of any service into the process, the orchestrator is responsible for the execution of the entire business process, and alternative scenarios can be planned when faults occur. For these reason we selected the orchestration approach for use in our prototype of the BMDS.

BPEL is similar in many respects to high-level programming languages. It has a number of basic activities that are functionally equvalent to typical programming constructs such as switch and conditional statements. BPEL allows basic activities such as invoking other WS, receiving requests from other services, conducting synchronous and asynchrounous operations, indicating faults, and waiting for specified periods of time. BPEL also supports the combination of the above activities and to defining more complex activities such as the sequencing of activities, invoking multiple services and activities in parrallel, executing loops, and selecting alternative paths or flows of execution. In addition, entire orchestrations can be encapsulated and deployed as a Web service in itself as shown in [39].

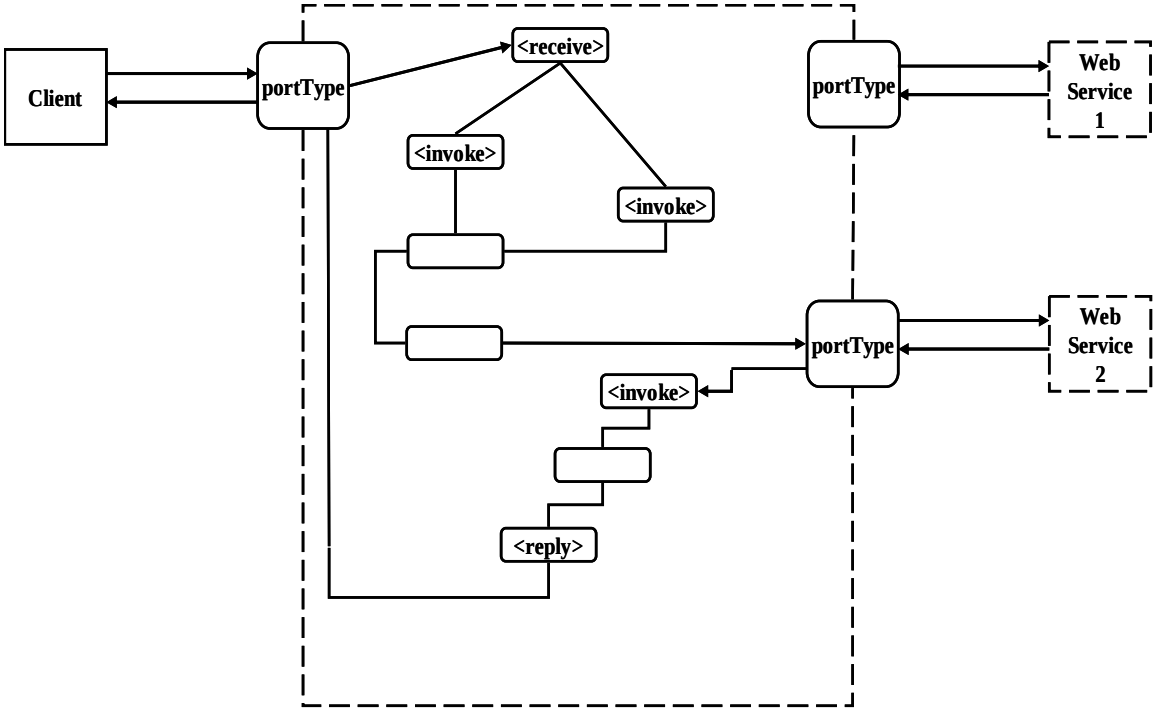

Figure 8. BPEL Process as a Web Service (After: [39])

Partner links are the connection between BPEL processes and WS. BPEL processes can invoke operations on other WS or BPEL processes can receive invocations from clients. In order for the invocation or the receive event to occur the portType shown above in Figure 8 must be exposed. The portType is associated with a specific partner link type, which declares how the two services will interact and what each service offers.

Finally, a partner link, of some partner link type, is created and becomes the concrete reference to the service that the BPEL service interacts with.

We have given a brief description of the core WS and BPEL standards. Each of the aforementioned standards is XML-based and falls somewhere into the WS technology stack shown below.

<table>
<tr><td colspan="8">BPEL</td></tr>
<tr><td colspan="2">WS-Atomic Transaction</td><td>WS-Business Activity</td><td rowspan="2">WS-Security</td><td>WS-Addressing</td><td>WS-Reliable Messaging</td><td>WS-Inspection</td><td>WS-Policy</td><td>WS-Eventing</td></tr>
<tr><td colspan="3">WS Coordination</td><td>WS-Addressing</td><td>WS-Reliable Messaging</td><td>WS-Inspection</td><td>WS-Policy</td><td>WS-Eventing</td></tr>
<tr><td colspan="8">Web Service Discovery (UDDI)</td></tr>
<tr><td colspan="8">Web Services (SOAP, WSDL)</td></tr>
<tr><td colspan="8">Specifications for Component Architecture (e.g. .Net/COM+ or Java EE)</td></tr>
</table>

Figure 9. WS Standards Stack (From: [39])

We expect that many of our services and orchestrations are necessarily transactional and must be secure. We do not, however, review these standards or investigate them for shortfalls, but present that as future work.

F. SERVICE ORIENTED ARCHITECTURE (SOA)

Service-Oriented Architecture (SOA): There is no single standard definition for SOA. Standards groups, information technology (IT) professionals, and SOA consultants, define SOA, in most cases, in slightly different ways [21][22][23]. While the definitions differ, they all have much in common. Any one of the definitions is sufficient for a basic understanding of what a SOA is all about.

In the most generic sense, a SOA is an architecture that supports the discovery of, binding to, and execution of some resource (service) or composition of resources (services) on a network. This type of architecture is commonly referred to as a distributed computing architecture and has been implemented in the past by such standards as Common Object Request Broker Architecture (CORBA) or by proprietary methods. The latest approach to SOA is one in which WS standards are used. There are advantages and disadvantages to all three of the approaches and no one approach is a "silver bullet" for distributed computing. However, the SOA with WS is currently receiving a significant amount of attention from the DoD and is the methodology the DoD has chosen for its GIG and NCES implementations. As such, we describe this SOA approach in the following section.

A SOA can be implemented using any number of existing technology standards from CORBA to WS. We make no argument for or against any approach. The DoD, however, has chosen SOA with WS and so we focus the next section on providing a basic understanding of SOA with WS. We use [21] as our primary reference for this section as it closely aligns with the DoD SOA vision and views WS as the technology that is currently most cable of implementing a SOA. In this section we summarize [21] and introduce its definitions of SOA, SOA characteristics, and service-orientation principles. We then provide an overview of the core standards of WS, followed by an overview of some of the existing extensions of the core set of WS standard. We conclude this section with a summary of the background information on SOA with WS.

SOA: [21] defines Contemporary SOA as follows:

> Contemporary SOA represents an open, agile, extensible, federated, composable architecture comprised of autonomous, QoS-capable, vendor diverse, interoperable, discoverable, and potentially reusable services, implemented as Web services.
>
> SOA can establish an abstraction of business logic and technology, resulting in loose coupling between these domains.

> SOA is an evolution of past platforms, preserving successful characteristics of traditional architectures, and bringing with it distinct principles that foster service-orientation in support of a service oriented enterprise.
>
> SOA is ideally standardized throughout an enterprise, but achieving this state requires a planned transition and support of a still evolving technology set.

The definition highlights many of the SOA characteristics and service-orientation principles that make SOA with WS an appealing choice for DoD. The definition is based on what [21] refers to as primitive SOA. Primitive SOA is based on the software engineering principle known as separation of concerns. Services encapsulate logic for solving the decomposed individual concerns of existing complex problems. How the services encapsulate logic, relate to each other, communicate, are designed and built are significant to how [21] defines SOA.

There are three basic components of the SOA architecture: service, description, and message. The service is the executable code; the (service) description contains the name of the service, location of the service, and the input and output exchange requirements; and messages are independent units of communication logic services use to communicate [21]. The components described above are generic enough to describe any distributed architecture. What makes SOA different is how each of these components is designed; using service-orientation principles.

Service-Orientation Principles: Service-orientation principles are "a set of principles most associated with service-orientation." These principles are applied to the design of each of the SOA components described above. The eight principles are listed in Table 1.

Service-orientation principle	***Brief description***
Services are reusable	Service are designed to support immediate and potential reuse
Services share a formal contract	Services are designed with formal contracts which describe the service and expose a

Service-orientation principle	***Brief description***
	services data sharing requirements
Services are loosely coupled	Services are designed to relate without dependencies on other services
Services abstract underlying logic	Service contracts are the only visible entity of a service. The actual service is of no concern to the user
Services are composable	Services can make up other services
Services are autonomous	Services are designed to be independent, self governing within an explicit boundary
Services are stateless	Services are designed so as not to manage state information
Services are discoverable	Services are designed to be discovered for use; they expose their formal contract for anyone to use

Table 1. Service-Orientation Principles (From: [21])

The principles above are applicable to the design and development of services. Thus far the definition of primitive SOA encompasses the core SOA components: service, description, and messages, and service-orientation principles. The final piece of the primitive SOA definition is what [21] calls the implementation platform that "pulls all of the pieces together to build a service-orientated solution" and the WS technology provides us that final piece.

Web services is playing a significant role in SOA these days and is best reflected by the following statement from [21].

> …the term "service-oriented" and various abstract SOA models existed before the arrival of Web services. However, no one technology advancement has been so suitable and successful in manifesting SOA than Web services.

The definition of contemporary SOA is based on primitive SOA. The important difference between the two is that primitive SOA represents what can and is currently being done with existing WS technology and contemporary SOA represents what is currently being done with current Web Services technology and what can be done in the future "with some extensions that rely on the availability of pre-defined Web services and corresponding vendor support."

Before discussing the WS technology we first highlight, in Table 2. below, a lists of common characteristics of contemporary SOA as identified in [21].

Common Characteristics of Contemporary SOA
Contemporary SOA is at the core of the service-oriented computing platform
Contemporary SOA increases Quality of Services (QoS)
Contemporary SOA is fundamentally autonomous
Contemporary SOA is based on open standards
Contemporary SOA supports vendor diversity
Contemporary SOA fosters intrinsic interoperability
Contemporary SOA promotes discovery
Contemporary SOA promotes federation
Contemporary SOA promotes architectural composability
Contemporary SOA fosters inherent reusability
Contemporary SOA emphasizes extensibility
Contemporary SOA supports a service-oriented business modeling paradigm
Contemporary SOA implements layers of abstraction
Contemporary SOA promotes loose coupling throughout the enterprise
Contemporary SOA promotes organizational agility
Contemporary SOA is a building block
Contemporary SOA is an evolution
Contemporary SOA is still maturing
Contemporary SOA is an achievable ideal

Table 2. Common characteristics of contemporary SOA (From: [21])

A full description of each characteristic can be found in [21]. We make reference to this list to make the point that contemporary SOA is not simply WS and service-orientation principles and that it is not something that is a finished product to be pulled off the shelf to provide a guaranteed solution. It is, as the characteristics show, an evolving and maturing concept. This work is about evolving and maturing SOA through the development of real-time choreographed WS extensions. Along that vein we now present a brief overview of each of the core WS standards: WSDL [14], SOAP [19] and UDDI [16], and one WS extension BPEL [15] as they are all essential in our work.

Web Services Description Language (WSDL), SOAP (formerly Simple Object Access Protocol), and Universal Description, Discovery, and Integration (UDDI) are the commonly referred to as the first generation WS standards. Each one of the standards represents a concern for the development of a WS. WSDL is a standard used to develop an XML based document that contains, at a minimum, the service name, location, and input and output requirements; this is the services contract. The WSDL document is a users' interface to an actual service and the information in a WSDL is captured in a UDDI registry so that potential users can "discover" the service and use it. UDDI is a specification used to design an xml-based registry service for Web services. The information contained in a WSDL is captured in a standard way, as outlined in the OASIS UDDI-specification, and mapped to a UDDI data model. In this way any registered service can be found. The SOAP standard describes the communications framework on which WS rests.

SOAP is an XML-based language and platform-independent communications protocol for exchanging messages between services over a network. A graphical representation of the core standards and how they relate is provided below in Figure 10.

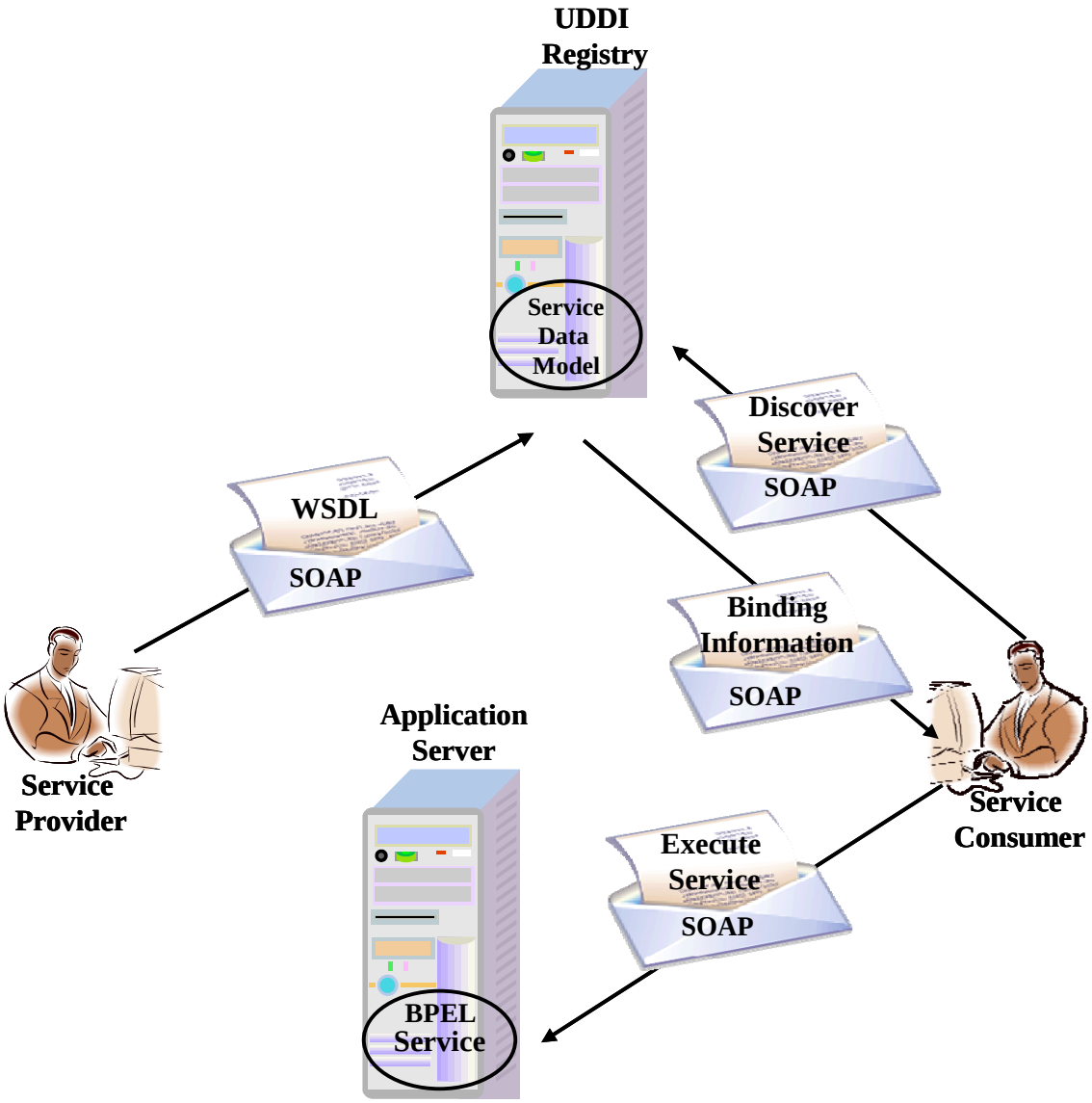

Figure 10. Basic SOA with core Web service standards

BPEL is not a core WS standard but it is, in our view, a standard that, with extensions, will be important for orchestrating services in a timely manner. In short,

orchestration describes how and when services of a business process interact. BPEL is a specific XML-based high-level language used to orchestrate WS for a particular process.

After review of each of the core WS and BPEL standards [14][15][16][17][18] we found that they lacked the constructs to deal with some of the timing deadlines that are intrinsic to many C4ISR systems in the DoD and therefore requires some extensions; this is the impetus for our research. We propose three contributions to advance the state of the art in SOA with WS: (i) extending the syntax of current standards, (ii) providing formal semantics for the extensions, and extending runtime support for our extensions.

III. COMMAND CONTROL AND BATTLE MANAGEMENT

A. INTRODUCTION

To remain competitive organizations have instituted business models and strategies to as rapidly as possible take advantage of advances in technology. [2] states that in the commercial sector, dominant competitors have developed information superiority and translated it into a competitive advantage by making the shift to network-centric operations.

The United States military is taking lessons learned from the commercial sector and paralleling that sector's efforts. Today's C4ISR systems and applications provide a good basis for achieving information superiority—possibly too good. General Tom Hobbins, Commander, U.S. Air Forces Europe writes in [35] that "Today's operator is drowning in information, yet starved for knowledge." The problem General Hobbins describes is that current C4ISR systems and applications do a good job at producing data, processing it, and moving it from location to location, but operators cannot then sift through the data and information fast enough to identify what information is necessary to increase a commanders knowledge base. Evidenced by General Hobbins quote above current technology is capable of enabling certain aspects of information superiority, but does not offer much in the way of helping operators or commanders to gain the necessary knowledge to make superior decisions.

This section of our research describes an approach to making the C4ISR information machine-processable using RDF to effectively process the masses of information created by C4ISR systems and applications. The rest of this chapter is organized as follows. In Section B we describe our approach. In Section C we model the knowledge captured in an OPORD using RDF. Section D describes how the OPORD, or any piece of it, can be written in message format to be exchanged between a hierarchical command structure of Battle Manager agents (commanders). In Section E we prototype the execution of the RDF OPORD by battle management systems using WS and specify a

C2 family of WS using WSDL and present their process integration and decision making capabilities using Business Process Execution Language.

B. APPROACH

Our approach to the problem described above is to relieve the operator and commander of some of the decision making burden, instead placing that burden on the information systems. To do this we must first design information systems to become knowledge aware in similar fashion to the Semantic Web. Specifically, we must take the documents, reports, and messages that convey important information to humans in military operations such as OPORDs, situation reports, status reports, etc. and equip computer systems with the ability to "understand" the who, what, where, when, and why of the information being exchanged. Computers cannot "understand" in the human sense, but they can be programmed to search, query, manipulate, and take action based on results by encoding these documents using RDF.

C. CREATING AN RDF OPORD

1. OPORD

The OPORD is the single most important document in any military operation. The OPORD is "a directive issued by a commander to subordinate commanders for the purpose of effecting the coordinated execution of an operation" [24]. It is a vital document in that it explains in detail the responsibilities of all participants in an operation. The OPORD shown in Figure 11 below, at a minimum, contains unit task organization and the five paragraphs of (1) Situation (2) Mission (3) Execution (4) and Service Support, and (5) Command and Signal. Figure 11 shows a standard Army OPORD format taken from [24].

[Classification]

[Change from verbal orders, if any]

Copy ## of ## copies
Issuing headquarters
Place of issue
Date-time group of signature
Message reference number

OPERATION PLAN/ORDER [number] [code name]
References
Time Zone Used Throughout the OPLAN/OPORD:
Task Organization
1. SITUATION.
a. Enemy forces.
b. Friendly forces.
c. Environment
(1). Terrain.
(2). Weather.
(3). Civil Considerations.
d. Attachments and detachments.
e. Assumptions.
2. MISSION.
3. EXECUTION.
Intent:
a. Concept of operations.
(1) Maneuver.
(2) Fires.
(3) Intelligence, Surveillance, and Reconnaissance.
(4) Intelligence.
(5) Engineer.
(6) Air and Missile Defense.
(7) Information Operations.
(8). Nuclear, Biological, Chemical.
(9). Military Police.
(10) Civil-Military Operations.
b. Tasks to maneuver units.
c. Tasks to other combat and combat support units.
d. Coordinating instructions.
(1) Time or condition when the plan/order becomes effective.
(2) CCIR (PIR, FFIR).
(3) Risk reduction control measures.
(4) Rules of engagement.
(5) Environmental considerations.
(6) Force protection.
(7) As required.
4. SERVICE SUPPORT (Support Concept).
b. Materiel and services.
c. Health service support.
d. Personnel.
e. As required.

5. COMMAND AND SIGNAL.
  a. Command.
  b. Signal.

ACKNOWLEDGE:
[Commander's last name]
[Commander's rank]
OFFICIAL:

[Authenticator's Name]
[Authenticator's Position]

ANNEXES:

DISTRIBUTION:

[Classification]

Figure 11. OPORD Format

In traditional land warfare combat, commanders issue their orders to subordinate commanders who in turn prepare and issue orders to their subordinates until each combatant in every unit knows his or her mission and the mission of those two levels up the chain of command. The initial OPORD of nearly all campaigns are routinely more detailed and well thought out than subsequent OPORDs. This tendency is a direct reflection of the amount of time available to plan prior to hostilities beginning. For the initial order, units may have days, weeks, and even months to plan and issue the orders. Once hostilities begin, the time to plan decreases and makes the development, issuance, and coordination of plans more difficult, in addition to reducing timelines to days or hours. In missile defense, the timelines are significantly shorter than traditional land warfare combat scenarios discussed above. For missile defense, timelines for engaging a threat missile can be on the order of seconds to several minutes.

With short timelines we look to perform autonomous execution of missile defense engagements where we remove the human from the loop. For this reason the OPORD must be designed to be read and "understood" by computers; we accomplish this by constructing OPORDs using RDF.

2. BMDS RDF Vocabulary

We briefly described RDF and RDF/XML previously in our background section we now apply it to our BMDS case study. The development of our vocabulary begins by identifying the specific kinds of resources we want to describe for our system. In the BMDS system we have many different orders and reports that must be exchanged and updated between commanders and subordinates. During hostilities, these commands and reports must be accurate and exchanged with all possible haste. In addition, we must be able to rapidly access and retrieve very specific parts of the orders and reports. The system must be able to be accessed by runtime applications and provide information about the resources to resource consumers. The information must be useful to both humans and automated processes.

Per [25] we define the BMDS domain elements and properties. As the BMDS system is large and complex we focus on the Operations Order as one web resource of many that might exist. The operations order in itself is a complex document that conveys a tremendous amount of knowledge as described previously in section 3.1. We borrow and modify some of the elements and properties listed in Table 3 from [25] as it is not practical to re-invent elements and properties if they already exist and work for the situation at hand. This is one benefit of using RDF: vocabularies are made available for reuse.

ELEMENT	PROPERTY	DESCRIPTION
Content	Unique Content ID	To identify content
	Biography	Content biographical information
	Relevancy	Relevancy of Content
	History	History of content movement
	Related	Related content
	Presentation	Content type and presentation

ELEMENT	PROPERTY	DESCRIPTION
Content bio	Title	Resources title
	Resource Abstract	Excerpt From resource if applicable
	Resource Description	Description of Resource
	Creation Date	Date Resource was first created
	Content Author	Person or Org responsible for creating content
	Content Owner	Person or Org who owns copyright on content
Relevancy	Content Status	Current status of content
	Subject	Subject/topic of resource (may duplicate)
	Relevancy Expiration	Date beyond which content is beyond useful
	References	External references
	Referenced by	External resources that reference content
History	Movement	Location at end of movement
	Reason	Reason for movement
	Date	Date of movement
	Type	Type of Movement
Related	Related Resource	Related Resource URI
	Reason	Reason for Relationship
Presentation	Format	Format of resource

ELEMENT	PROPERTY	DESCRIPTION
	Conformity	Standards/specs resource conforms to (may repeat)
	Requires	Resource dependencies (may repeat)

Table 3. Elements and Properties (From: [25])

In addition, we introduce a number of properties that identify the paragraphs and sub-paragraphs of the standard five-paragraph OPORD. We designed these paragraphs and sub-paragraphs as separate HTML pages so that they can be developed and processed separately. In Listings Listing 3 and Table 4 below we show the OPORD header in RDF/XML and the BMDS RDF Schema respectively for the Resource Class and the biography property.

```
1: <?xml version="1.0"?>
2: <rdf:RDF xml:lang="en"
3: xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:  xmlns:bmds="http://swe.nps.edu/bmds/elements/1.0/"
5:  xmlns:geo=
"http://www.w3.org/2003/01/geo/wgs84_pos#"
6:  xmlns:dcterms="http://purl.org/dc/terms"
7:  xmlns:dc="http://purl.org/dc/elements/1.1/"
8:  xml:base="http://swe.nps.edu/bmds/opords/">
9:
10: <bmds:Resource rdf:about="sca120080129.htm">
11: <!--Resource biographical information-->
13: <bmds:bio rdf:parseType="Resource">
14:  <dc:title>opord sca120080129</dc:title>
15:<dc:dateCreated>2008-01-29T00:00:00
19:00</dc:dateCreated>
16: <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
17:  <dc:author>COL Smith</dc:author>
18: <bmds:issueHq>STRATCOM</bmds:issueHq>
19:<bmds:classification>Unclassified</bmds:classification>
20:   <bmds:placeIssued>
21:      <geo:Point>
22:        <geo:lat>20.20</geo:lat>
23:        <geo:long>-90.80</geo:long>
24:       </geo:Point>
25:   </bmds:placeIssued>
26:<bmds:msgRefNum>Message Reference Number
</bmds:msgRefNum>
```

```
27:     <bmds:orderNum>OPORD NUM</bmds:orderNum>
28:       <bmds:codeName>BUTKUS</bmds:codeName>
29:       <bmds:cdrLname>COOK</bmds:cdrLname>
30:       <bmds:cdrRank>GEN</bmds:cdrRank>
31:   </bmds:bio>
```

Listing 3. OPORD Header

```
1: <?xml version="1.0"?>
2: <rdf:RDF xml:lang="en"
3: xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
5:<rdfs:Class rdf:about="http://swe.nps.edu/bmds/elements/1.0/Resource">
6 :
7 :
8:<rdfs:isDefinedBy rdf:resource="http://swe.nps.edu/bmds/elements/1.0/"/>
9:<rdfs:label xml:lang="en"> BMDS Network Resource</rdfs:label>
10:<rdfs:comment xml:lang="en">
11:   BMDS network resource managed by bmds system
12:</rdfs:comment>
13:  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource" />
14: </rdfs:Class>
15:<rdfs:Property rdf:about="http://swe.nps.edu/bmds/elements/1.0/bio">
18:<rdfs:isDefinedByrdf:resource="http://swe.nps.edu/bmds/elements/1.0/"/>
19:<rdfs:label xml:lang="en">Resource biography</rdfs:label>
20:    <rdfs:comment xml:lang="en">
21:      Biographical information for resource
22:    </rdfs:comment>
23:<rdf:range rdf:resource="http://swe.nps.edu/bmds/elements/1.0/Resource" />
24:<rdf:domainrdf:resource="http://swe.nps.edu/bmds/elements/1.0/Resource" />
25:  </rdfs:Property>
26: </rdf:RDF>
```

Listing 4. BMDS RDF Schema

Listing 3 shows the RDF/XML model instance used to validate the BMDS RDFS shown in Figure 4. We begin with Listing 3 at line 1 with a standard XML declaration followed in line 2 by the initial rdf:RDF tag, that sets the language to “en” for English. We note here the facility for multi-language support—a necessity in multi-national coalition operations.

Lines 3-7 identify the namespaces used in our application and their associated prefixes. Specifically, DC is the namespace prefix for the Dublin Core metadata element set used in cross-domain information resource description, geo is the namespace prefix

for w3c metadata element set used in to represent spatial information such as latitude and longitude. "bmds" is the prefix we have chosen for our namespace to represent information within the ballistic missile defense domain. Line 8 identifies the base namespace used throughout the document allowing us to identify resources using only the element name as shown in line 10 (e.g. sca120080129.htm). Line 10 defines sca120080129.htm as a bmds:Resource; the Operations Order that is "issued" or "processed " by humans or machines. Looking at Listing 4, line 5 we show the main object of our document, "Resource", is defined as an RDF Class in our schema http://swe.nps.edu/bmds/elements/1.0/ . This is a subclass of the RDF Resource type.

Going back to Listing 3 we show that Line 9 provides a label for human consumption and defines the Resource as a "BMDS Network resource. "Resource" turns out to be the only rdf:Class in this particular effort, the rest of the vocabulary elements are properties. The properties include data type information, labels, and describe the relationship between the properties and the classes. For the sake of brevity, we only describe the schema bio property in Listing 4 that defines the biographical property of sca120080129.htm Resource shown in Listing 3 from lines 11-31. Specifically, we highlight, in [25] parlance, that in the schema the rdfs:domain property, associates the bio property with the resource it modifies; Resource (sca120080129.htm). In other words, the domain for the bio property are those elements defined in the schema of type Resource. In a like manner the bio element has a range that can only contain Resource types.

The above paragraph provides a small example of how using RDF/XML syntax we can describe our domain to produce the RDF and RDFS documents. The rest of the domain elements are described in a similar manner. A critical recurring step that takes place during the development of the RDF and RDFS is checking the validity of the RDF/XML specification.

In this effort after adding each new resource, property, or class, we checked our RDF and RDFS using the World Wide Web Validation Service provided at http://www.w3.org/RDF/Validator/ . The service, as the name indicates, validates that the RDF/XML code submitted is syntactically correct. If the code validates the service

provides a 3-tuple representation of the data model shown below in Table 4 and its associated graphical representation shown in Figure 12, Figure 13, and Figure 14.

Num	Subject	Predicate	Object
1	http://swe.nps.edu/bmds/opords/sca120080129.htm	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://swe.nps.edu/bmds/elements/1.0/Resource
2	http://swe.nps.edu/bmds/opords/sca120080129.htm	http://swe.nps.edu/bmds/elements/1.0/bio	genid:A259319
3	genid:A259319	http://purl.org/dc/elements/1.1/title	"opord sca120080129"@en
4	genid:A259319	http://purl.org/dc/elements/1.1/dateCreated	"2008-01-29T00:00:00-19:00"@en
5	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/timeZoneUsed	"ZULU"@en
6	genid:A259319	http://purl.org/dc/elements/1.1/author	"COL Smith"@en
7	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/issueHq	"STRATCOM"@en
8	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/classification	"Unclassified"@en
9	genid:A259320	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2003/01/geo/wgs84_pos#Point
10	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/placeIssued	genid:A259320
11	genid:A259320	http://www.w3.org/2003/01/geo/wgs84_pos#lat	"20.20"@en
12	genid:A259320	http://www.w3.org/2003/01/geo/wgs84_pos#long	"-90.80"@en
13	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/msgRefNum	"MessageReferenceNumber"@en
14	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/orderNum	"OPORD NUM"@en
15	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/codeName	"BUTKUS"@en
16	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/cdrLname	"COOK"@en
17	genid:A259319	http://swe.nps.edu/bmds/elements/1.0/cdrRank	"GEN"@en

Table 4. BMDS SCA OPORD 3-tuples

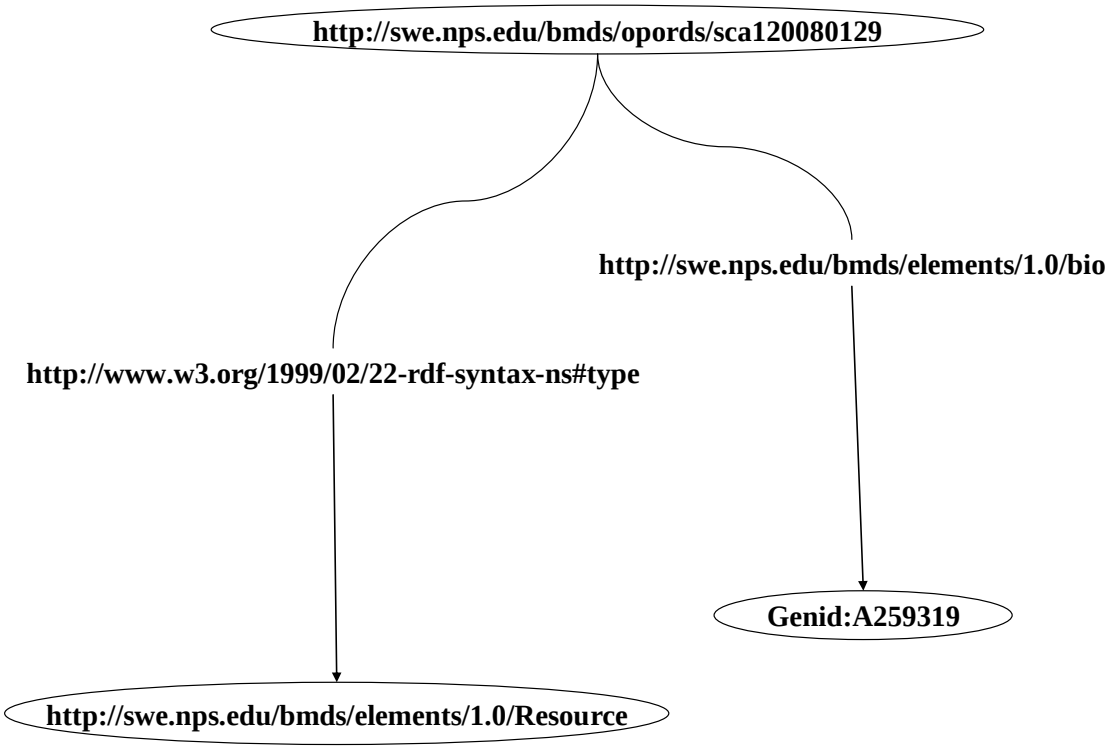

Figure 12. SCA1 OPORD directed Graph part-1

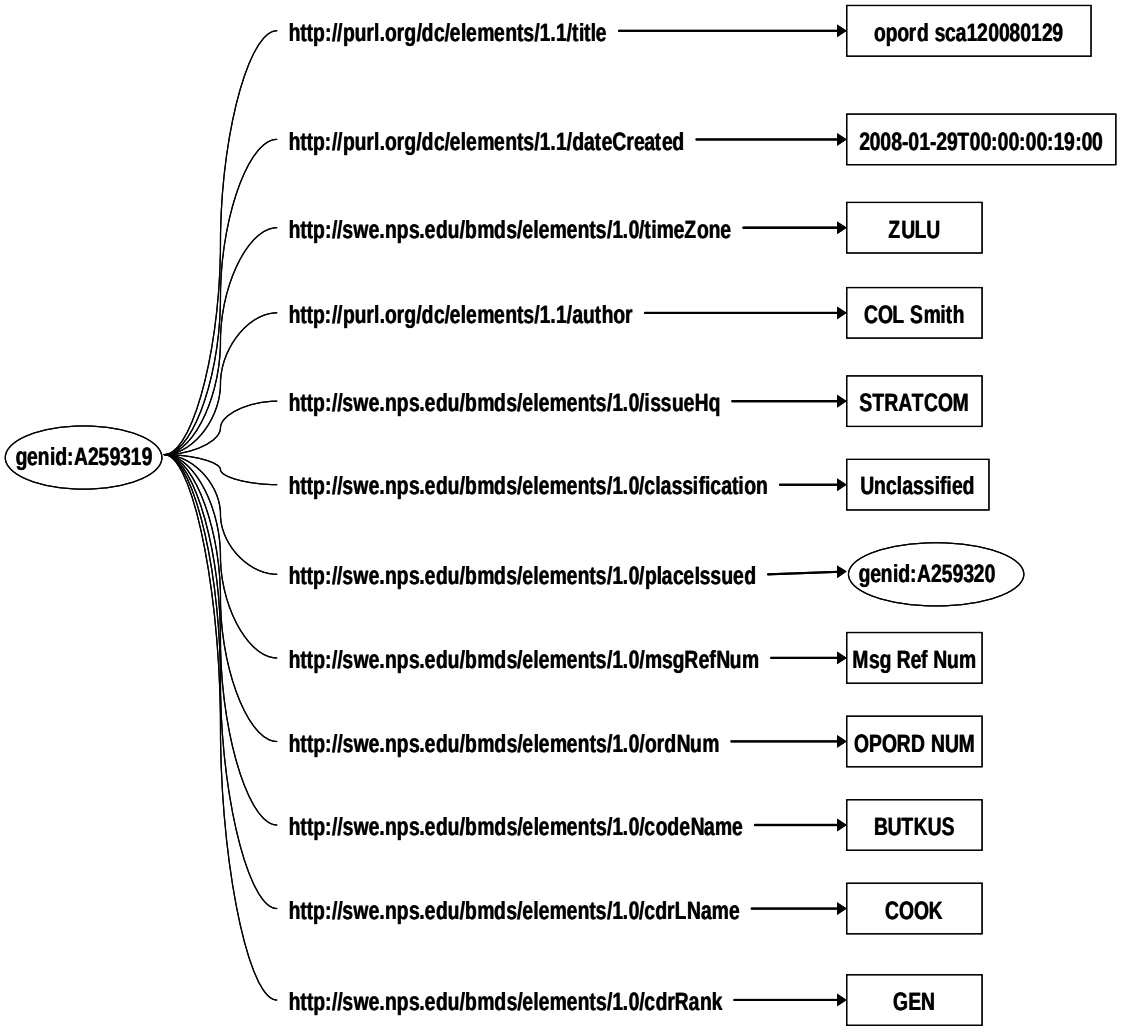

Figure 13. SCA1 OPORD directed Graph part-2

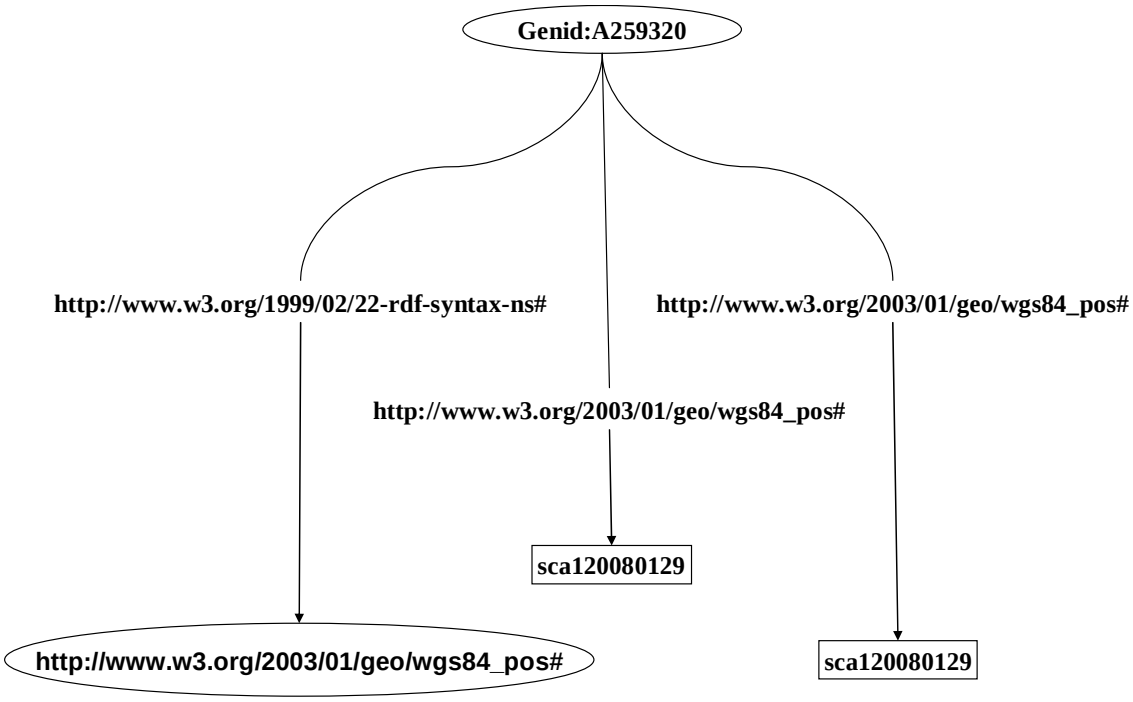

Figure 14. SCA1 OPORD directed Graph part-3

Thus far we have shown small RDF/XML code snippets from our RDF OPORD and its associated RDF Schema (vocabulary). We have shown how we went about developing these pieces. Using the manner described above we developed four OPORDs, all of which appear in Appendix A. In the next section, we discuss the OPORD in terms of a hypothetical scenario and show the importance of the orders being machine-readable.

D. DEVELOPING AND ISSUING RDF OPORDS

OPORDs are developed at the highest level of command and subsequently issued to the next lower level of command. The receiving command's staff processes the higher level command's order and develops its own order and issues that modified order to all of its subordinates. This sequence continues down the chain of command until every soldier in has received an order. As the OPORD passes down the chain some of the paragraphs and sub-paragraphs are re-written or modified while others remain unchanged. Over the next several sub-sections we will breakdown and explain in detail each piece of the

operations orders we have developed. We show, using a hypothetical scenario, how the operations order is developed at the top-level command and flows down to the lowest level command.

1. Scenario

Our hypothetical scenario proceeds as follows. Strategic Command Agent (SCA1) deploys its subordinate command, Regional Command Agent 2's (RCA), as depicted in Figure 15. RCA2's organic sensor and Battle Manager are deployed to protect the SCA1 western flank from missile attacks. RCA2 assigns its subordinate command TCA21 the responsibility of defending the country of Mainland's coast and assets along its coast and TCA22 the responsibility of defending the island and its local assets. The SCA Intel officer estimates that the enemies likely air avenue of attack comes from the North East, depicted in Figure 15 by the large dotted arrow and labeled as such. In similar fashion each subordinate TCA processes the higher level order and writes and issues its own order; deploying their weapons and sensor systems also shown below in Figure 15. We move now into descriptions of the various portions of the SCA1 operations order.

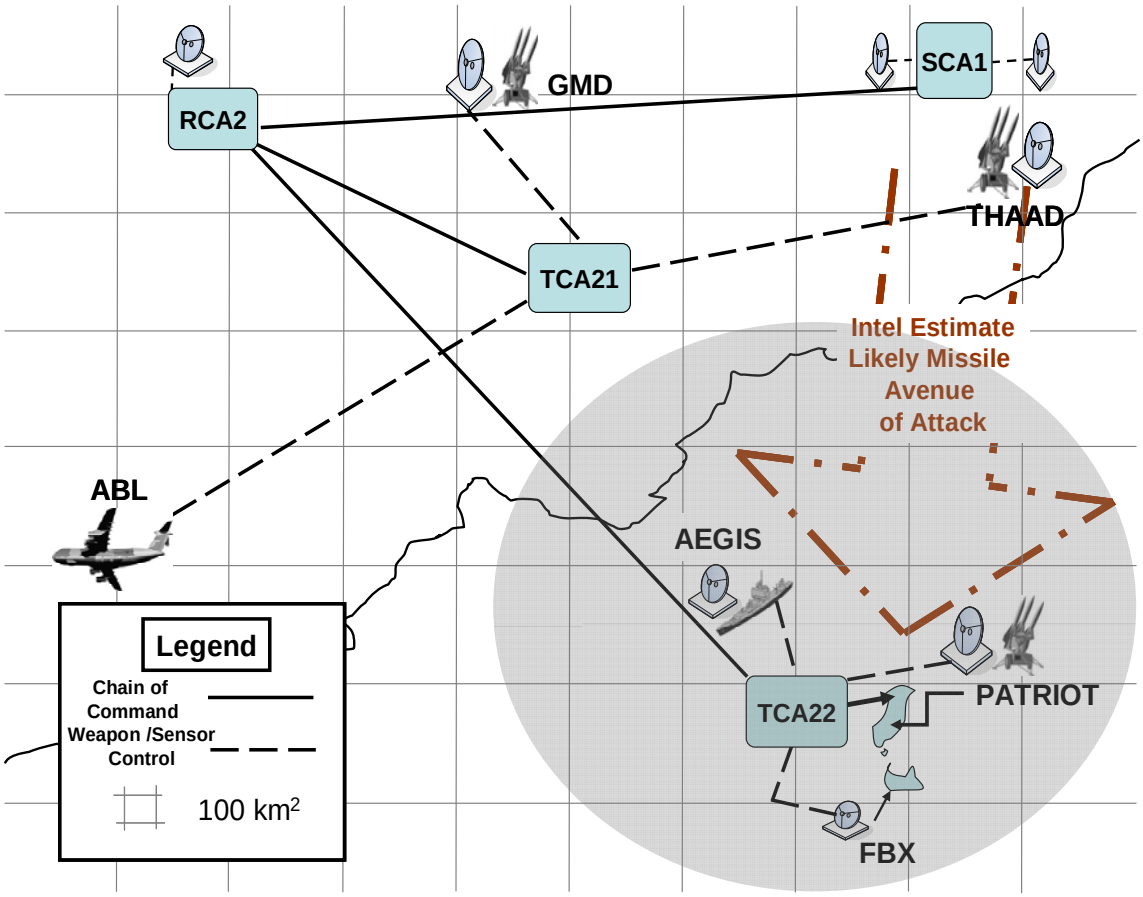

Figure 15. OPORD Intel Estimate

2. OPORD Header

In Listing 3 we described some of the RDF/XML syntax necessary to serialize our OPORD and provided a brief description of the namespaces. In this section we will describe the domain specific information contained in the header and show how it relates back to the OPORD Format shown above in Figure 11. In Listing 3 lines 13-31 describe the bio element of the resource defined at line 10; the sca120080129.htm operations order. The bio element provides biographical information about the identified resource that in our case is the SCA1 operations order. Line 14 identifies "opord sca120080129" as the dc:title of our resource. We use dc:title in this case as it provides a ready made standard to describe the title of the SCA1 operations order, which maps to the Operation Plan/Order Number code in OPORD Format. The date-time group signature of the OPORD is represented in RDF/XML at line 15; it is defined by the dc:dateCreated tag

and represents the effective time for implementing the order. Line 16 shows that the time zone used throughout this order is Zulu, that is, Greenwich Mean Time (GMT). Line 17 contains the last name and rank of the author preparing the order. Line 18 shows the responsible headquarters for issuing the order. Line 19 shows the classification of the operations order; typically placed at the top and bottom of every page of the order. Lines 20-25 identify the location of the issuing headquarters. The location is represented using the geo namespace which offers a standard means of representing spatial information and in our case gives a point location for the issuing headquarters. Line 26 contains information for identifying the Message Reference Number. This number is issued by the unit signal officer and is used by addressees to acknowledge receiving the order over non secure communications networks. Lines 27-28 contain the operations orders number and code name. Lines 29-30 contain the last name and rank of the commanding officer of the issuing unit.

The next section of the OPORD header contains information on the relevancy of the document, the name and position of the authenticating authority for the order, and how and what is needed to properly present and consume the resource by its recipients. We show in Listing 5 at lines 47-56 that this order's presentation format is in the form of text and HTML and that it conforms to XHTML 1.0 and CSS standards, and requires an HTML user agent. Lines 57-61 show that the resource requires special parsing of type logo and the actual value to be parsed is http://swe.nps.edu/logo/nps.jpg.

```
33:   <!--Resource's Relevancy at time RDF/XML document was built-->
34:   <bmds:relevancy rdf:parseType="Resource">
35:     <bmds:currentStatus>Active</bmds:currentStatus>
36:     <dcterms:valid >2008-01-29T00:00:00-19:00</dcterms:valid>
37:     <dcterms:references rdf:resource="http://swe.nps.edu/bmds/oplan/lambert.htm" />
38:     <dcterms:isReferencedBy> </dcterms:isReferencedBy>
39:   </bmds:relevancy>
40:
41: <!--Resource's Authenticator-->
42:   <bmds:authenticator rdf:parseType="Resource">
43:     <bmds:authenName>PULFORD</bmds:authenName>
44:     <bmds:authenPos>J3</bmds:authenPos>
45:   </bmds:authenticator>
46:
47: <!--Resource's presentation/consumption information about resource-->
48:   <bmds:presentation rdf:parseType="Resource">
49:       <dc:format>text/html</dc:format>
50:       <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
51:       <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
52:       <dcterms:requires>HTML User agent</dcterms:requires>
53:       <bmds:requires rdf:parseType="Resource">
54:         <bmds:type>stylesheet</bmds:type>
55:         <rdf:value>http://swe.nps.edu/opord.css</rdf:value>
56:       </bmds:requires>
57:       <bmds:requires rdf:parseType="Resource">
58:         <bmds:type>logo</bmds:type>
59:         <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
60:       </bmds:requires>
61:   </bmds:presentation>
```

Listing 5. OPORD Header continued

The bio element, as described in Listing 5, provides information about the OPORD document. We continue the use of the bio element in subsequent paragraphs of the OPORD so that they can be developed separately and still be associated with the OPORD.

Next, we look at the final piece of the main OPORD. This part identifies the resources that are internal to the OPORD. They identify the five paragraphs of the OPORD and all of its associated annexes.

```
<!--Resources internal to opord SCA120080129-->
67:   <bmds:related rdf:resource="situation.htm" />
68:   <bmds:related rdf:resource="mission.htm" />
69:   <bmds:related rdf:resource="execution.htm" />
70:   <bmds:related rdf:resource="serviceSpt.htm" />
71:   <bmds:related rdf:resource="cmdSig.htm" />
72:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexATaskO.htm" />
73:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexBIntel.htm" />
74:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexCOps.htm" />
75:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexDLog.htm" />
76:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexEPers.htm" />
77:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexFPubAffair.htm" />
78:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexGCivAffair.htm" />
79:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexHMetoc.htm" />
80:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexJCmdRel.htm" />
81:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexKC3.htm" />
82:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexLEnviron.htm" />
83:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexMMapChrtGeo.htm" />
84:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexNSpaceOps.htm" />
85:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexPHostNatSpt.htm" />
86:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexQMedServ.htm" />
87:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexSSpecTechOps.htm" />
88:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexTConsMgt.htm" />
89:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexVIntAgentCoord.htm" />
90:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexXExeChkList.htm" />
91:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/annexs/annexZDistro.htm" />
```

Listing 6. Main OPORD paragraphs and annexes

In Listing 6, lines 67-71 identify each of the five paragraphs of the OPORD as their own separate resources related to OPORD SCA120080129. Lines 72-91 identify all of the Annexes associated with the OPORD. Because these paragraphs and annexes are their own resources they can be written in the same way, using RDF/XML, that the main OPORD has been written. We present the separate paragraphs in order.

3. OPORD Situation

```
<!--RELATED RESOURCES-->
97:
98: <!--PARAGRAPH 1 SITUATION-->
99: <bmds:Resource rdf:about="situation.htm">
100:
101: <!--Resource biographical information-->
102:   <bmds:bio rdf:parseType="Resource">
103:      <dc:title>SITUATION</dc:title>
104:      <dc:dateCreated>2008-01-29T00:00:00-18:00</dc:dateCreated>
105:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
106:      <dc:author>LTC Brown</dc:author>
107:      <bmds:issueHq>STRATCOM</bmds:issueHq>
108:      <bmds:classification>Unclassified</bmds:classification>
109:      <bmds:placeIssued>
110:        <geo:Point>
111:          <geo:lat>20.20</geo:lat>
112:          <geo:long>-90.80</geo:long>
113:        </geo:Point>
114:      </bmds:placeIssued>
115:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
116:      <bmds:orderNum>OPORD NUM</bmds:orderNum>
117:      <bmds:codeName>BUTKUS</bmds:codeName>
118:      <bmds:cdrLname>COOK</bmds:cdrLname>
119:      <bmds:cdrRank>GEN</bmds:cdrRank>
120:   </bmds:bio>
121:
122:   <!--Resource's Relevancy at time RDF/XML document was built-->
123:   <bmds:relevancy rdf:parseType="Resource">
124:     <bmds:currentStatus>Active</bmds:currentStatus>
125:     <dcterms:valid >2008-01-29T00:00:00-18:00</dcterms:valid>
126:   </bmds:relevancy>
127:
128: <!--Resource's Authenticator-->
129:   <bmds:authenticator rdf:parseType="Resource">
130:     <bmds:authenName>MILLER</bmds:authenName>
131:     <bmds:authenPos>Asst J3</bmds:authenPos>
132:   </bmds:authenticator>
133:
134: <!--Resource's presentation/consumption information about resource-->
135:   <bmds:presentation rdf:parseType="Resource">
136:      <dc:format>text/html</dc:format>
137:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
138:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
139:      <dcterms:requires>HTML User agent</dcterms:requires>
140:      <bmds:requires rdf:parseType="Resource">
141:        <bmds:type>stylesheet</bmds:type>
142:        <rdf:value>http://swe.nps.edu/sit.css</rdf:value>
143:      </bmds:requires>
144:      <bmds:requires rdf:parseType="Resource">
```

```
145:        <bmds:type>logo</bmds:type>
146:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
147:      </bmds:requires>
148:   </bmds:presentation>
149:
150: <!--Resources internal to SITUATION-->
151:   <bmds:related rdf:resource="enemyForces.htm" />
152:   <bmds:related rdf:resource="friendlyForces.htm" />
153:   <bmds:related rdf:resource="environment.htm" />
154:   <bmds:related rdf:resource="attachDetach.htm" />
155:   <bmds:related rdf:resource="assumptions.htm" />
156:
157: </bmds:Resource>
```

Listing 7. OPORD Situation Paragraph

We note first in the situation paragraph, as in all of the others, it begins with the dc:bio tag and contains all of the same type of information as the main OPORD. The difference between this paragraph and the main OPORD with respect to the bio element is that the values of the element tags sometimes differ (e.g. the author for this particular paragraph is LTC Brown where as the author for the entire OPORD is COL Smith). Line 150 shows that the situation paragraph, like the main OPORD, has some internal resources. These resources are the sub-paragraphs titled Enemy Forces and Friendly Forces and are shown below in Listing 8.

```
163: <bmds:Resource rdf:about="enemyForces.htm">
164:
165:
166: <!--Resource biographical information-->
167:   <bmds:bio rdf:parseType="Resource">
168:      <dc:title>ENEMY FORCES</dc:title>
169:      <dc:dateCreated>2008-01-29T00:00:00-17:21</dc:dateCreated>
170:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
171:      <dc:author>LTC Brooks</dc:author>
172:      <bmds:issueHq>STRATCOM</bmds:issueHq>
173:      <bmds:classification>Unclassified</bmds:classification>
174:      <bmds:placeIssued>
175:        <geo:Point>
176:          <geo:lat>20.20</geo:lat>
177:          <geo:long>-90.80</geo:long>
178:        </geo:Point>
179:      </bmds:placeIssued>
180:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
181:      <bmds:orderNum>OPORD NUM</bmds:orderNum>
182:      <bmds:codeName>BUTKUS</bmds:codeName>
```

```
183:      <bmds:cdrLname>COOK</bmds:cdrLname>
184:      <bmds:cdrRank>GEN</bmds:cdrRank>
185:   </bmds:bio>
186:
187:   <!--Resource's Relevancy at time RDF/XML document was built-->
188:   <bmds:relevancy rdf:parseType="Resource">
189:     <bmds:currentStatus>Active</bmds:currentStatus>
190:     <dcterms:valid >2008-01-29T00:00:00-17:21</dcterms:valid>
191:   </bmds:relevancy>
192:
193: <!--Resource's Authenticator-->
194:   <bmds:authenticator rdf:parseType="Resource">
195:     <bmds:authenName>MILLER</bmds:authenName>
196:     <bmds:authenPos>Asst J3</bmds:authenPos>
197:   </bmds:authenticator>
198:
199: <!--Resource's presentation/consumption information about resource-->
200:   <bmds:presentation rdf:parseType="Resource">
201:      <dc:format>text/html</dc:format>
202:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
203:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
204:      <dcterms:requires>HTML User agent</dcterms:requires>
205:      <bmds:requires rdf:parseType="Resource">
206:        <bmds:type>stylesheet</bmds:type>
207:        <rdf:value>http://swe.nps.edu/sit.css</rdf:value>
208:      </bmds:requires>
209:      <bmds:requires rdf:parseType="Resource">
210:        <bmds:type>logo</bmds:type>
211:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
212:      </bmds:requires>
213:   </bmds:presentation>
214:
215: <!--Resources internal to Enemy Forces-->
216:   <bmds:related rdf:resource="elcoa.htm" />
217:   <bmds:related rdf:resource="emdcoa.htm" />
218:   <bmds:related rdf:resource="elcoaSketch.htm" />
219:   <bmds:related rdf:resource="emdcoaSketch.htm" />
```

Listing 8. Enemy Forces

Again, we note the bio element from lines 167-185 and that lines 215-219 identify the internal resources related to the Enemy Forces paragraph. Let us now look at the internal resources of the enemy forces sub-paragraph represented in Listing 9 elcoa.htm and elcoaSketch.htm and in Listing 10 emdcoa.htm and emdcoaSketch.htm

```
.224: <!--RELATED Enemy Forces RESOURCES-->
225: <bmds:Resource rdf:about="elcoa.htm">
226: <!--Resource biographical information-->
227:   <bmds:bio rdf:parseType="Resource">
228:      <dc:title>Enemy Likely COA </dc:title>
229:      <dc:dateCreated>2008-01-29T00:00:00-17:21</dc:dateCreated>
230:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
231:      <dc:author>LTC Brooks</dc:author>
232:      <bmds:issueHq>STRATCOM</bmds:issueHq>
233:      <bmds:classification>Unclassified</bmds:classification>
234:      <bmds:placeIssued>
235:        <geo:Point>
236:          <geo:lat>20.20</geo:lat>
237:          <geo:long>-90.80</geo:long>
238:        </geo:Point>
239:      </bmds:placeIssued>
240:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
241:      <bmds:orderNum>OPORD NUM</bmds:orderNum>
242:      <bmds:codeName>BUTKUS</bmds:codeName>
243:      <bmds:cdrLname>COOK</bmds:cdrLname>
244:      <bmds:cdrRank>GEN</bmds:cdrRank>
245:   </bmds:bio>
246:
247:   <!--Resource's Relevancy at time RDF/XML document was built-->
248:   <bmds:relevancy rdf:parseType="Resource">
249:     <bmds:currentStatus>Active</bmds:currentStatus>
250:     <dcterms:valid >2008-01-29T00:00:00-17:21</dcterms:valid>
251:   </bmds:relevancy>
252:
253: <!--Resource's Authenticator-->
254:   <bmds:authenticator rdf:parseType="Resource">
255:     <bmds:authenName>MILLER</bmds:authenName>
256:     <bmds:authenPos>Asst J3</bmds:authenPos>
257:   </bmds:authenticator>
258:
259: <!--Resource's presentation/consumption information about resource-->
260:   <bmds:presentation rdf:parseType="Resource">
261:      <dc:format>text/html</dc:format>
262:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
263:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
264:      <dcterms:requires>HTML User agent</dcterms:requires>
265:      <bmds:requires rdf:parseType="Resource">
266:        <bmds:type>stylesheet</bmds:type>
267:        <rdf:value>http://swe.nps.edu/enemyForces.css</rdf:value>
268:      </bmds:requires>
269:      <bmds:requires rdf:parseType="Resource">
270:        <bmds:type>logo</bmds:type>
271:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
272:      </bmds:requires>
273:   </bmds:presentation>
274:
275: <bmds:elcoa>
```

```
276:    <rdf:Seq>
277:      <rdf:_1>
278:        <geo:Point>
279:         <geo:lat>43.79</geo:lat>
280:         <geo:long>73.19</geo:long>
281:        </geo:Point>
282:      </rdf:_1>
283:      <rdf:_2>
284:        <geo:Point>
285:         <geo:lat>42.45</geo:lat>
286:         <geo:long>73.48</geo:long>
287:        </geo:Point>
288:      </rdf:_2>
289:      <rdf:_3>
290:        <geo:Point>
291:         <geo:lat>38.51</geo:lat>
292:         <geo:long>77.02</geo:long>
293:        </geo:Point>
294:      </rdf:_3>
295:      <rdf:_4>
296:        <geo:Point>
297:         <geo:lat>35.10</geo:lat>
298:         <geo:long>79.01</geo:long>
299:        </geo:Point>
300:      </rdf:_4>
301:      <rdf:_5>
302:       <geo:Point>
303:         <geo:lat>25.48</geo:lat>
304:         <geo:long>80.16</geo:long>
305:       </geo:Point>
306:      </rdf:_5>
307:    </rdf:Seq>
308:  </bmds:elcoa>
309:
310: </bmds:Resource>
311:
312: <!--RELATED Enemy Forces RESOURCES-->
313: <bmds:Resource rdf:about="elcoaSketch.htm">
314: <!--Resource biographical information-->
315:  <bmds:bio rdf:parseType="Resource">
316:     <dc:title>Enemy Likely COA Sketch</dc:title>
317:     <dc:dateCreated>2008-01-29T00:00:00-17:21</dc:dateCreated>
318:     <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
319:     <dc:author>LTC Brooks</dc:author>
320:     <bmds:issueHq>STRATCOM</bmds:issueHq>
321:     <bmds:classification>Unclassified</bmds:classification>
322:     <bmds:placeIssued>
323:       <geo:Point>
324:         <geo:lat>20.20</geo:lat>
325:         <geo:long>-90.80</geo:long>
326:       </geo:Point>
327:     </bmds:placeIssued>
```

```
328:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
329:      <bmds:orderNum>OPORD NUM</bmds:orderNum>
330:      <bmds:codeName>BUTKUS</bmds:codeName>
331:      <bmds:cdrLname>COOK</bmds:cdrLname>
332:      <bmds:cdrRank>GEN</bmds:cdrRank>
333:   </bmds:bio>
334:
335:   <!--Resource's Relevancy at time RDF/XML document was built-->
336:   <bmds:relevancy rdf:parseType="Resource">
337:     <bmds:currentStatus>Active</bmds:currentStatus>
338:     <dcterms:valid >2008-01-29T00:00:00-17:21</dcterms:valid>
339:   </bmds:relevancy>
340:
341: <!--Resource's Authenticator-->
342:   <bmds:authenticator rdf:parseType="Resource">
343:     <bmds:authenName>MILLER</bmds:authenName>
344:     <bmds:authenPos>Asst J3</bmds:authenPos>
345:   </bmds:authenticator>
346:
347: <!--Resource's presentation/consumption information about resource-->
348:   <bmds:presentation rdf:parseType="Resource">
349:      <dc:format>text/html</dc:format>
350:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
351:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
352:      <dcterms:requires>HTML User agent</dcterms:requires>
353:      <bmds:requires rdf:parseType="Resource">
354:        <bmds:type>stylesheet</bmds:type>
355:        <rdf:value>http://swe.nps.edu/enemyForces.css</rdf:value>
356:      </bmds:requires>
357:      <bmds:requires rdf:parseType="Resource">
358:        <bmds:type>logo</bmds:type>
359:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
360:      </bmds:requires>
361:   </bmds:presentation>
362:
363: </bmds:Resource>
364:
```

Listing 9. Enemy Forces Most Likely Course Of Action

The information in Listing 9 and Listing 10 are near identical. As such we will describe both in this paragraph. Listing 9, lines 224–245, contains the bio information for the enemy's most likely course of action (emlcoa) resource. Listing 10 shows the information for the enemy's most dangerous course of action (emdcoa) (lines 365-386).

The next twenty-eight lines in each of the listings contains the relevancy information, resource authenticator, presentation and consumption information for the

respective resources. Lines 335-339 in Listing 9 and lines 388-392 in Listing 10 describe the relevancy information—the fact that they are both valid and active resources beginning at 17:21 hours on January 1, 2008. Lines 341- 346 in Listing 9 and lines 394-399 in Listing 10 describe the resource authenticators last name as Miller whose position is the J3 (Global Operations Officer). Lines 347-363 in Listing 9 and lines 400-415 in Listing 10 contain the requirements for the resources to be displayed and consumed for and by users.

Listing 9, lines 275-308, identifies the enemy's most likely air avenue of approach. Listing 10, lines 416-449, describes the enemy's most dangerous air avenue of approach. The avenues of approach are described using the WGS84 Geo spatial namespace from the Semantic Web interest group. The points defined in the avenues of approach are representative of what might be sent from a higher level command to that command's subordinate command. We will see later how the receiving command further refines the avenues of approach and passes that information down to it subordinates.

The final section of both Listing 9 and Listing 10, beginning at line 313 and line 454 respectively, are the description of the sketches used to portray the avenues of approach.

```
<!--RELATED Enemy Forces RESOURCES-->
<bmds:Resource rdf:about="emdcoa.htm">
<!--Resource biographical information-->
  <bmds:bio rdf:parseType="Resource">
     <dc:title>Enemy Most Dangerous COA </dc:title>
     <dc:dateCreated>2008-01-29T00:00:00-17:21</dc:dateCreated>
     <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
     <dc:author>LTC Brooks</dc:author>
     <bmds:issueHq>STRATCOM</bmds:issueHq>
     <bmds:classification>Unclassified</bmds:classification>
     <bmds:placeIssued>
       <geo:Point>
         <geo:lat>20.20</geo:lat>
         <geo:long>-90.80</geo:long>
       </geo:Point>
     </bmds:placeIssued>
     <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
     <bmds:orderNum>OPORD NUM</bmds:orderNum>
     <bmds:codeName>BUTKUS</bmds:codeName>
     <bmds:cdrLname>COOK</bmds:cdrLname>
     <bmds:cdrRank>GEN</bmds:cdrRank>
  </bmds:bio>

  <!--Resource's Relevancy at time RDF/XML document was built-->
  <bmds:relevancy rdf:parseType="Resource">
    <bmds:currentStatus>Active</bmds:currentStatus>
    <dcterms:valid >2008-01-29T00:00:00-17:21</dcterms:valid>
  </bmds:relevancy>

<!--Resource's Authenticator-->
  <bmds:authenticator rdf:parseType="Resource">
    <bmds:authenName>MILLER</bmds:authenName>
    <bmds:authenPos>Asst J3</bmds:authenPos>
  </bmds:authenticator>

<!--Resource's presentation/consumption information about resource-->
  <bmds:presentation rdf:parseType="Resource">
     <dc:format>text/html</dc:format>
     <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
     <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
     <dcterms:requires>HTML User agent</dcterms:requires>
     <bmds:requires rdf:parseType="Resource">
       <bmds:type>stylesheet</bmds:type>
       <rdf:value>http://swe.nps.edu/enemyForces.css</rdf:value>
     </bmds:requires>
     <bmds:requires rdf:parseType="Resource">
       <bmds:type>logo</bmds:type>
       <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
     </bmds:requires>
  </bmds:presentation>

<bmds:emdcoa>
```

```
417:    <rdf:Seq>
418:      <rdf:_1>
419:        <geo:Point>
420:         <geo:lat>52.20</geo:lat>
421:         <geo:long>92.80</geo:long>
422:        </geo:Point>
423:      </rdf:_1>
424:      <rdf:_2>
425:        <geo:Point>
426:         <geo:lat>52.01</geo:lat>
427:         <geo:long>91.30</geo:long>
428:        </geo:Point>
429:      </rdf:_2>
430:      <rdf:_3>
431:        <geo:Point>
432:         <geo:lat>45.20</geo:lat>
433:         <geo:long>91.80</geo:long>
434:        </geo:Point>
435:      </rdf:_3>
436:      <rdf:_4>
437:        <geo:Point>
438:         <geo:lat>43.05</geo:lat>
439:         <geo:long>90.80</geo:long>
440:        </geo:Point>
441:      </rdf:_4>
442:      <rdf:_5>
443:        <geo:Point>
444:         <geo:lat>39.22</geo:lat>
445:         <geo:long>89.80</geo:long>
446:        </geo:Point>
447:      </rdf:_5>
448:    </rdf:Seq>
449:  </bmds:emdcoa>
450:
451: </bmds:Resource>
452:
453: <!--RELATED Enemy Forces RESOURCES-->
454: <bmds:Resource rdf:about="emdcoaSketch.htm">
455: <!--Resource biographical information-->
456:  <bmds:bio rdf:parseType="Resource">
457:     <dc:title>Enemy Most Dangerous COA Sketch</dc:title>
458:     <dc:dateCreated>2008-01-29T00:00:00-17:21</dc:dateCreated>
459:     <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
460:     <dc:author>LTC Brooks</dc:author>
461:     <bmds:issueHq>STRATCOM</bmds:issueHq>
462:     <bmds:classification>Unclassified</bmds:classification>
463:     <bmds:placeIssued>
464:       <geo:Point>
465:         <geo:lat>20.20</geo:lat>
466:         <geo:long>-90.80</geo:long>
467:       </geo:Point>
468:     </bmds:placeIssued>
```

```
469:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
470:      <bmds:orderNum>OPORD NUM</bmds:orderNum>
471:      <bmds:codeName>BUTKUS</bmds:codeName>
472:      <bmds:cdrLname>COOK</bmds:cdrLname>
473:      <bmds:cdrRank>GEN</bmds:cdrRank>
474:   </bmds:bio>
475:
476:   <!--Resource's Relevancy at time RDF/XML document was built-->
477:   <bmds:relevancy rdf:parseType="Resource">
478:     <bmds:currentStatus>Active</bmds:currentStatus>
479:     <dcterms:valid >2008-01-29T00:00:00-17:21</dcterms:valid>
480:   </bmds:relevancy>
481:
482: <!--Resource's Authenticator-->
483:   <bmds:authenticator rdf:parseType="Resource">
484:     <bmds:authenName>MILLER</bmds:authenName>
485:     <bmds:authenPos>Asst J3</bmds:authenPos>
486:   </bmds:authenticator>
487:
488: <!--Resource's presentation/consumption information about resource-->
489:   <bmds:presentation rdf:parseType="Resource">
490:      <dc:format>text/html</dc:format>
491:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
492:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
493:      <dcterms:requires>HTML User agent</dcterms:requires>
494:      <bmds:requires rdf:parseType="Resource">
495:        <bmds:type>stylesheet</bmds:type>
496:        <rdf:value>http://swe.nps.edu/enemyForces.css</rdf:value>
497:      </bmds:requires>
498:      <bmds:requires rdf:parseType="Resource">
499:        <bmds:type>logo</bmds:type>
500:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
501:      </bmds:requires>
502:   </bmds:presentation>
503:
504: </bmds:Resource>
505:
506:
507: <!--END SITUATION,ENEMY FORCES-->
```

Listing 10. Enemy Forces Most Dangerous Course Of Action

We next describe the Friendly Forces sub-paragraph of the Situation shown below in Listing 11. This paragraph states the mission, commander's intent, and concept of operations (CONOPS) for the headquarters one and two levels up. In the case of the SCA there is no higher headquarters so we adjust the OPORD accordingly by not showing this part of the paragraph. The next portion of the paragraph is the resources internal to the friendly forces paragraph shown from the sketch of the friendly forces at

line 565 to the terrain, weather, and the civil construction shown in lines 736, 737, and 738 respectively. In this order the terrain and civil considerations are listed as “NONE,” shown at lines 798 and line 920. However the weather, for the general area of operations is listed in lines 805-865 and play a significant role in ballistic missile defense. As the order flows down the chain of command, the weather, like most other areas of the order, becomes more detailed.

511: <!--SITUATION,FRIENDLY FORCES-->
512:
513: <bmds:Resource rdf:about="friendlyForces.htm">

. bio
. relevancy
. authenticator
. presentation/Consumption

565: <!--Resources internal to FRIENDLY FORCES-->
573: <bmds:Resource rdf:about="friendlyForcesSketch.htm">
. bio
. relevancy
. authenticator
. presentation/Consumption
734:
735: <!--Resources internal to Enemy Forces-->
736: <bmds:related rdf:resource="terrain.htm" />
737: <bmds:related rdf:resource="weather.htm" />
738: <bmds:related rdf:resource="civilConsideration.htm" />
. bio
. relevancy
. authenticator
. presentation/Consumption
797:
798: <bmds:terrain>NONE</bmds:terrain>
799:
800: </bmds:Resource>
801:
802: <!--RELATED RESOURCES-->
803: <!--SITUATION,ENVIRONMENT, Weather-->
804:
805: <bmds:Resource rdf:about="weather.htm">
. bio
. relevancy
. authenticator
. presentation/Consumption
856:

```
857:   <bmds:weather>
858:     <bmds:tempF>98</bmds:temp>
859:     <bmds:baroPressure>28.89</bmds:baroPressure>
860:     <bmds:windKnots>3</bmds:windKnots>
861:<bmds:windDir>NE</windDir>
862:     <bmds:visibility>100</bmds:visibility>
863:     <bmds:precipitation>0</bmds:precipitation>
864:   </bmds:weather>
870: <bmds:Resource rdf:about="civilConsideration.htm">
```

. bio
. relevancy
. authenticator
. presentation/Consumption

```
919:
920:   <bmds:civilCons>NONE</bmds:civilCons>
923: </bmds:Resource>
924:
925: <!--END FRIENDLY FORCES-->
926: <!--END RELATED RESOURCES-->
927: <!--END PARAGRAPH 1 SITUATION-->
```

Listing 11. Friendly Forces

This wraps up the situation paragraph for the highest level command, SCA1. We next show the SCA1 command's mission statement and how it is written in RDF/XML.

The Mission statement is one of the most critical paragraphs of the OPORD, it is typically written out in paragraph form to address the who, what, where, when, and why of a mission. In this work we chose to capture the who, what, where, when, and why more formally by using the tags bmds:who, bmds:what, bmds;where, bmds:when, and bmds:why. This provides a means of making the paragraph machine-reader friendly. We explain the code in Listing 12 in more detail.

We begin at line 988 and 989 by identifying the mission paragraph of SCA1.htm as a resource. The mission statement is further broken down into the bmds:who, bmds:what, bmds;where, bmds:when, and bmds:why sections as discussed above. Line 989 bmds:who that mission is assigned to the resource SCA1, the bmds:what section of the mission paragraph is defined by the resource http://swe.nps.edu/bmds/ujtl/st6-1-5.htm at line 990 which identifies a specific missile-defense task from the Universal Joint Task List. Line 991 shows the PDAL.htm resource, that lists, in priority order, all of the assets to be defended by SCA1, further, each member of the list has its own machine-readable

self-describing document, as we will discuss later. Line 992 defines the time at which the defense of all assets is to be established. The bmd:why element of the mission is at line 993 and provides a simple description of why the mission is being performed.

The internal resources portion of the mission statement begins at line 999 and identifies the Priority Defended Asset List PDAL.htm as the resource about to be described. Lines 1000 to 1056 are used to describe the PDAL document and its state. Starting at line 1057 we show the list of defended assets by marking them inside the RDF sequence tag and giving each asset a sequence number using the RDF format "rdf_#" where the # indicates the sequence and priority number of the asset. As an example line 1060 taken from Listing 12 looks as follows.

1060: <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/Island_X.htm"/>

It identifies Island X as the number one priority for SCA1 to defend. In the same vein line 1086 taken from Listing 12 shown below identifies West Land Stadium as the 27th priority or the lowest priority to defend for SCA1.

1086: <rdf:_27
rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandStadium.htm"/>

All other assets, in priority order, are listed in Listing 12 from line 1061 - 1085.

4. OPORD Mission

```
987:  <!--OPORD MISSION STATEMENT-->
988:   <bmds:mission rdf:parseType="Resource">
989:     <bmds:who rdf:resource="http://swe.nps.edu/bmds/units/SCA1.htm" />
990:     <bmds:what rdf:resource="http://swe.nps.edu/bmds/ujtl/st6-1-5.htm" />
991:     <bmds:where rdf:resource="PDAL.HTML" />
992:     <bmds:when>2008-02-23T00:00:00-04:00</bmds:when>
993:     <bmds:why>Preserve Peace</bmds:why>
994:   </bmds:mission>
995:
996:
997:
998: <!--Resources internal to mission-->
999:   <bmds:related rdf:resource="PDAL.htm" />
1000:
1001: </bmds:Resource>
```

```
1002:
1003:
1004: <bmds:Resource rdf:about="PDAL.htm">
1005:
1006:   <!--Resource biographical information-->
1007:   <bmds:bio rdf:parseType="Resource">
1008:       <dc:title>PDAL</dc:title>
1009:       <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
1010:       <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
1011:       <dc:author>COL Smith</dc:author>
1012:       <bmds:issueHq>STRATCOM</bmds:issueHq>
1013:       <bmds:classification>Unclassified</bmds:classification>
1014:       <bmds:placeIssued>
1015:         <geo:Point>
1016:           <geo:lat>20.20</geo:lat>
1017:           <geo:long>-90.80</geo:long>
1018:         </geo:Point>
1019:       </bmds:placeIssued>
1020:       <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
1021:       <bmds:orderNum>OPORD NUM</bmds:orderNum>
1022:       <bmds:codeName>BUTKUS</bmds:codeName>
1023:       <bmds:cdrLname>COOK</bmds:cdrLname>
1024:       <bmds:cdrRank>GEN</bmds:cdrRank>
1025:   </bmds:bio>
1026:
1027:   <!--Resource's Relevancy at time RDF/XML document was built-->
1028:   <bmds:relevancy rdf:parseType="Resource">
1029:     <bmds:currentStatus>Active</bmds:currentStatus>
1030:     <dcterms:valid >2008-01-29T00:00:00-19:00</dcterms:valid>
1031:     <dcterms:references rdf:resource="http://swe.nps.edu/bmds/oplan/lambert.htm" />
1032:     <dcterms:isReferencedBy> </dcterms:isReferencedBy>
1033:   </bmds:relevancy>
1034:
1035: <!--Resource's Authenticator-->
1036:   <bmds:authenticator rdf:parseType="Resource">
1037:     <bmds:authenName>PULFORD</bmds:authenName>
1038:     <bmds:authenPos>J3</bmds:authenPos>
1039:   </bmds:authenticator>
1040:
1041: <!--Resource's presentation/consumption information about resource-->
1042:   <bmds:presentation rdf:parseType="Resource">
1043:       <dc:format>text/html</dc:format>
1044:       <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
1045:       <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
1046:       <dcterms:requires>HTML User agent</dcterms:requires>
1047:       <bmds:requires rdf:parseType="Resource">
1048:         <bmds:type>stylesheet</bmds:type>
1049:         <rdf:value>http://swe.nps.edu/opord.css</rdf:value>
1050:       </bmds:requires>
1051:       <bmds:requires rdf:parseType="Resource">
1052:         <bmds:type>logo</bmds:type>
1053:         <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
```

```
1054:       </bmds:requires>
1055:   </bmds:presentation>
1056:
1057: <!--OPORD PDAL-->
1058:   <bmds:pdal>
1059:     <rdf:Seq>
1060:       <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/Island_X.htm"/>
1061:       <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm"/>
1062:       <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm"/>
1063:       <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXAirPort.htm"/>
1064:       <rdf:_5 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXShippingPort.htm"/>
1065:       <rdf:_6 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYMinOfDef.htm"/>
1066:       <rdf:_7 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCommCntr.htm"/>
1067:       <rdf:_8 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYWtrTreatFac.htm"/>
1068:       <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCapitalCity.htm"/>
1069:       <rdf:_10 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandSecurity.htm"/>
1070:       <rdf:_11 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandCapitalBldg.htm"/>
1071:       <rdf:_12 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandPowerPlant.htm"/>
1072:       <rdf:_13 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandAirPort.htm"/>
1073:       <rdf:_14 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandTransport.htm"/>
1074:       <rdf:_15 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandMinOfDef.htm"/>
1075:       <rdf:_16 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandCommCtr.htm"/>
1076:       <rdf:_17 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandWtrTreatFac.htm"/>
1077:       <rdf:_18 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandStadium.htm"/>
1078:       <rdf:_19 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandSecurity.htm"/>
1079:       <rdf:_20 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandCapBuilding.htm"/>
1080:       <rdf:_21 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandPowerPlant.htm"/>
1081:       <rdf:_22 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandAirPort.htm"/>
1082:       <rdf:_23 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandTransport.htm"/>
1083:       <rdf:_24 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandMinOfDef.htm"/>
1084:       <rdf:_25 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandCommunicationsCenter"/>
1085:       <rdf:_26 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandWtrTreatFac.htm"/>
1086:       <rdf:_27 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandStadium.htm"/>
1087:     </rdf:Seq>
1088:   </bmds:pdal>
```

Listing 12. OPORD Mission statement

Lines 1091 through 1119 simply identify the UJTL task st6-1-5.htm and each separate PDAL asset as a separate resource related to the mission document. This is done because these individual documents contain a large amount of essential information describing the title source. We remove most of the code to save space, but provide an example at line 1094. Line 1094 identifies http://swe.nps.edu/bmds/PDAL/ IslXCap_Building.htm as a resource describing Island X's capitol building. The actual description for the capitol building is shown in lines 1136-1144. This provides

information such as the assets center of mass location and what type of asset it is (e.g., the capitol is a building made of stone). This format is followed for every asset on the defended asset list (PDAL.htm).

```
1089:
1090:
1091: <!--Resources internal to mission-->
1092:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/ujtl/st6-1-5.htm" />
1093:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/PDAL/Island_X.htm"/>
1094:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm"/>
.
.
.
1118:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandWtrTreatFac.htm"/>
1119:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/
PDAL/WLandStadium.htm"/>
1120:
1121:
1122:
1123: </bmds:Resource>
1124:
1125:
1126: <rdf:Description rdf:about="http://swe.nps.edu/bmds/PDAL/Island_X.htm">
1127:    <bmds:location>
1128:     <geo:Point>
1129:       <geo:lat>62.20</geo:lat>
1130:         <geo:long>80.80</geo:long>
1131:       </geo:Point>
1132:    </bmds:location>
1133:   <bmds:assetType>land mass</bmds:assetType>
1134: </rdf:Description>
1135:
1136: <rdf:Description rdf:about="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm">
1137:    <bmds:location>
1138:     <geo:Point>
1139:       <geo:lat>58.20</geo:lat>
1140:         <geo:long>78.80</geo:long>
1141:       </geo:Point>
1142:    </bmds:location>
1143:  <bmds:assetType>Stone Building</bmds:assetType>
1144: </rdf:Description>
1145:
1146: <rdf:Description rdf:about="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm">
1147:   <bmds:location>
1148:     <geo:Point>
1149:       <geo:lat>67.20</geo:lat>
1150:         <geo:long>60.80</geo:long>
1151:       </geo:Point>
1152:    </bmds:location>
```

```
1153:   <bmds:assetType>Combined Cycle Plant</bmds:assetType>
1154: </rdf:Description>
1155:
.
.
.
1376:
1377: <rdf:Description rdf:about="http://swe.nps.edu/bmds/PDAL/WestLandStadium">
1378:  <bmds:location>
1379:    <geo:Point>
1380:      <geo:lat>80.20</geo:lat>
1381:        <geo:long>70.80</geo:long>
1382:      </geo:Point>
1383:   </bmds:location>
1384:   <bmds:assetType>National Memorial Stadium</bmds:assetType>
1385: </rdf:Description>
1386:
```

Listing 13. OPORD Priority Defended Asset List and Measures of Effectiveness

Looking back to line 1092 of Listing 13 that line identifies http://swe.nps.edu/bmds/ujtl/st6-1-5.htm as an internal resource to the mission resource document. The information captured in that resource is identified in Listing 13 starting at line 1388 where http://swe.nps.edu/bmds/ujtl/st6-1-5.htm is referenced as the resource being described. Line 1389 identifies the particular task as Organize and Coordinate Theater Missile Defense. We use this as a representative task only. The SCA would typically describe and assign strategic level missions and tasks to its units. Next, starting at line 1390, we add to the OPORD some important information for our research. We add a list of MOEs. These particular MOEs come from the UJTL and help us track how effective the unit and equipment are in accomplishing the task identified earlier as 'Organize and Coordinate Theater Missile Defense". In our OPORD we show ten MOEs from line 1392 through 1408. These MOE are further defined in Listing 14 lines 1416 through 1503. We remove some of the descriptions of the MOEs to save space, but provide a few examples. Line 1417 defines the measure as casualties per day and line 1418 describes the measure description as the casualties per day attributed to enemy missile attacks. Another example at lines 1463-1466 show the measure as percentage and the description as the percentage of protected Defended Asset List (DAL) locations, successfully defended.

```
1387:
1388: <rdf:Description rdf:about="http://swe.nps.edu/bmds/ujtl/st6-1-5.htm">
1389:  <bmds:task>Organize and Coordinate Theater Missile Defense</bmds:task>
1390:   <bmds:MOE>
1391:    <rdf:Seq>
1392:     <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M1.htm"/>
1393:     <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M2.htm"/>
1394:     <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M3.htm"/>
1395:     <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M4.htm"/>
1396:     <rdf:_5 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M5.htm"/>
1397:     <rdf:_6 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M6.htm"/>
1398:     <rdf:_7 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M7.htm"/>
1399:     <rdf:_8 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M8.htm"/>
1400:     <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M9.htm"/>
1401:     <rdf:_10 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M10.htm"/>
1402:     <rdf:_11 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M11.htm"/>
1403:     <rdf:_12 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M12.htm"/>
1404:     <rdf:_13 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M13.htm"/>
1405:     <rdf:_14 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M14.htm"/>
1406:     <rdf:_15 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M15.htm"/>
1407:     <rdf:_17 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M17.htm"/>
1408:     <rdf:_18 rdf:resource="http://swe.nps.edu/bmds/UJTL/ST6-1-5M18.htm"/>
1409:    </rdf:Seq>
1410:  </bmds:MOE>
1411: </rdf:Description>
1412:
1413:
1414:
1415:
1416: <rdf:Description rdf:about="http://swe.nps.edu/bmds/UJTL/ST6-1-5M1.htm">
1417: <bmds:measure>CasualtiesPerDay</bmds:measure>
1418: <bmds:measureDescr>Attributed to enemy
missile attacks (host-nation civilian)
</bmds:measureDescr>
1419: </rdf:Description>
1420:
.
.
.
1452:
1453: <rdf:Description rdf:about="http://swe.nps.edu/bmds/UJTL/ST6-1-5M8.htm">
1454: <bmds:measure>Percent</bmds:measure>
1455: <bmds:measureDescr>Of launched ballistic
missiles, destroyed before impact
</bmds:measureDescr>
1456: </rdf:Description>
1457:
1458: <rdf:Description rdf:about="http://swe.nps.edu/bmds/UJTL/ST6-1-5M9.htm">
1459: <bmds:measure>Percent</bmds:measure>
1460: <bmds:measureDescr>Of launched cruise missiles,
destroyed before impact.</bmds:measureDescr>
```

```
1461: </rdf:Description>
1462:
1463: <rdf:Description rdf:about="http://swe.nps.edu/bmds/UJTL/ST6-1-5M10.htm">
1464: <bmds:measure>Percent</bmds:measure>
1465: <bmds:measureDescr>Of protected DAL
locations, successfully defended.
</bmds:measureDescr>
1466: </rdf:Description>
1467:
.
.
.
1499: <rdf:Description rdf:about="http://swe.nps.edu/bmds/UJTL/ST6-1-5M18.htm">
1500: <bmds:measure>Minute</bmds:measure>
1501: <bmds:measureDescr>From detection/
identification of TM elements to ordinance
release against validated TM
1502: target</bmds:measureDescr>
1503: </rdf:Description>
1504:
1505:
1506: <!--END PARAGRAPH 2 MISSION-->
```

Listing 14. Measures of Effectiveness

This completes the mission statement of the OPORD. We move now to the execution paragraph of the OPORD and detail the specifics in the next paragraph.

5. OPORD Execution

The execution paragraph of the OPORD contains a large amount of important information. The execution document attributes are identified in Listing 15 in lines 1514-1577. The first piece of the execution paragraph is the commander's intent; Per [24] the commander's intent is "a clear, concise statement of what the force must do and the conditions the force must meet to succeed with respect to the enemy, terrain, and the desired end state." The commander's intent is captured as an internal resource to the execution paragraph in our order at line 1570. The definition for the commander's intent begins at line 1582 but the actual content is not defined until line 1635 of Listing 15 .

In the same manner, we write the concept of the operation, identifing it as a resource at line 1571 in Listing 15; the definition starts at line 1642 and the content is written at line 1695 where the concept reads as "defend per the PDAL".

```
1514: <!--OPORD EXECUTION STATEMENT-->
1515:  <bmds:Resource rdf:about="execution.htm">
1516:
1517:   <!--Resource biographical information-->
1518:   <bmds:bio rdf:parseType="Resource">
1519:      <dc:title>EXECUTION</dc:title>
1520:      <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
1521:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
1522:      <dc:author>COL Smith</dc:author>
1523:      <bmds:issueHq>STRATCOM</bmds:issueHq>
.
.
.
1553:   <bmds:presentation rdf:parseType="Resource">
1554:      <dc:format>text/html</dc:format>
1555:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
1556:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
1557:      <dcterms:requires>HTML User agent</dcterms:requires>
1558:      <bmds:requires rdf:parseType="Resource">
1559:        <bmds:type>stylesheet</bmds:type>
.
.
.
1567:
1568:
1569: <!--Resources internal to opord EXECUTION-->
1570:   <bmds:related rdf:resource="cdrsIntent.htm" />
1571:   <bmds:related rdf:resource="conceptOfOps.htm" />
1572:   <bmds:related rdf:resource="taskToSubs.htm" />
1573:   <bmds:related rdf:resource="taskToSptUnit.htm" />
1574:
1575: </bmds:Resource>
1576:
1577: <!--RELATED RESOURCES-->
1578:
1579:
1580:
1581: <!--OPORD EXECUTION, COMMANDERS INTENT -->
1582:  <bmds:Resource rdf:about="cdrsIntent.htm">
1583:
1584:   <!--Resource biographical information-->
1585:   <bmds:bio rdf:parseType="Resource">
1586:      <dc:title>CDR INTENT</dc:title>
1587:      <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
1588:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
1589:      <dc:author>COL Smith</dc:author>
1590:      <bmds:issueHq>STRATCOM</bmds:issueHq>
.
.
.
1634:
```

```
1635:  <bmds:cdrsIntent>Destroy ballistic missiles prior to keep out range</bmds:cdrsIntent>
1636:
1637:
1638: </bmds:Resource>
1639:
1640:
1641: <!--OPORD EXECUTION, CONCEPTS OF OPERATIONS -->
1642:  <bmds:Resource rdf:about="conceptOfOps.htm">
1643:
1644:   <!--Resource biographical information-->
1645:   <bmds:bio rdf:parseType="Resource">
1646:       <dc:title>CONCEPTS OF OPERATIONS</dc:title>
1647:       <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
1648:       <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
1649:       <dc:author>COL Smith</dc:author>
1650:       <bmds:issueHq>STRATCOM</bmds:issueHq>
.
.
.
1694:
1695: <bmds:conceptsOfOps>Defend per the PDAL
</bmds:conceptsOfOps>
1696:
1697: </bmds:Resource>
1698:
1699:
1700:
```

Listing 15. OPORD Execution Paragraph

The next two sections of the execution paragraph are the tasks assigned to maneuver units and tasked to other combat and combat support units. In this case there are no other units so we address only SCA1 units. The tasks to the subordinate units definition starts at line 1702 in Listing 16 by identifying taskToSubs.htm as the resource being described using the rdf:about tag. We identify in the tasks to subordinates list the Regional Command Agent (RCA) followed by those assets that the command is responsible for defending. In line 1757 we identify RCA2 as the command responsible for defending the assets listed from lines 1759 to line 1770. The assets are listed in priority using the rdf:_# tag and so rdf:_1 http://swe.nps.edu/bmds/PDAL/Island_X.htm is assigned to RCA2 and it is first on RCA2's priority list while

http://swe.nps.edu/bmds/PDAL/MnLandYCapitalCity.htm at line 1768 is RCA2's ninth and final priority. The task to subordinates RCA1 and RCA3 are listed in priority order at lines 1773 and 1788 respectively.

```
1701: <!--OPORD EXECUTION, TASKS TO SUBORDINATES -->
1702:  <bmds:Resource rdf:about="taskToSubs.htm">
1703:
1704:   <!--Resource biographical information-->
1705:   <bmds:bio rdf:parseType="Resource">
1706:      <dc:title>TASK TO SUBORDINATES</dc:title>
1707:      <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
1708:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
1709:      <dc:author>COL Smith</dc:author>
1710:      <bmds:issueHq>STRATCOM</bmds:issueHq>
1711:      <bmds:classification>Unclassified</bmds:classification>
1712:      <bmds:placeIssued>
1713:        <geo:Point>
1714:          <geo:lat>20.20</geo:lat>
1715:          <geo:long>-90.80</geo:long>
1716:        </geo:Point>
1717:      </bmds:placeIssued>
1718:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
1719:      <bmds:orderNum>OPORD NUM</bmds:orderNum>
1720:      <bmds:codeName>BUTKUS</bmds:codeName>
1721:      <bmds:cdrLname>COOK</bmds:cdrLname>
1722:      <bmds:cdrRank>GEN</bmds:cdrRank>
1723:   </bmds:bio>
1724:
1725:   <!--Resource's Relevancy at time RDF/XML document was built-->
1726:   <bmds:relevancy rdf:parseType="Resource">
1727:     <bmds:currentStatus>Active</bmds:currentStatus>
1728:     <dcterms:valid >2008-01-29T00:00:00-19:00</dcterms:valid>
1729:     <dcterms:references rdf:resource="http://swe.nps.edu/bmds/oplan/lambert.htm" />
1730:     <dcterms:isReferencedBy> </dcterms:isReferencedBy>
1731:   </bmds:relevancy>
1732:
1733: <!--Resource's Authenticator-->
1734:   <bmds:authenticator rdf:parseType="Resource">
1735:     <bmds:authenName>PULFORD</bmds:authenName>
1736:     <bmds:authenPos>J3</bmds:authenPos>
1737:   </bmds:authenticator>
1738:
1739: <!--Resource's presentation/consumption information about resource-->
1740:   <bmds:presentation rdf:parseType="Resource">
1741:      <dc:format>text/html</dc:format>
1742:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
1743:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
1744:      <dcterms:requires>HTML User agent</dcterms:requires>
1745:      <bmds:requires rdf:parseType="Resource">
```

```
1746:       <bmds:type>stylesheet</bmds:type>
1747:       <rdf:value>http://swe.nps.edu/taskToSubs.css</rdf:value>
1748:     </bmds:requires>
1749:     <bmds:requires rdf:parseType="Resource">
1750:       <bmds:type>logo</bmds:type>
1751:       <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
1752:     </bmds:requires>
1753:   </bmds:presentation>
1754:
1755: <bmds:taskToSub rdf:parseType="Resource">
1756:
1757:   <bmds:unit rdf:resource="http://swe.nps.edu/bmds/agent/RCA2.htm"/>
1758:     <bmds:defends>
1759:     <rdf:Seq>
1760:      <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/Island_X.htm"/>
1761:      <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm"/>
1762:      <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm"/>
1763:      <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXAirPort.htm"/>
1764:      <rdf:_5 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXShippingPort.htm"/>
1765:      <rdf:_6 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYMinOfDef.htm"/>
1766:      <rdf:_7 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCommCntr.htm"/>
1767:      <rdf:_8 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYWtrTreatFac.htm"/>
1768:      <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCapitalCity.htm"/>
1769:    </rdf:Seq>
1770:   </bmds:defends>
1771:
1772:
1773:   <bmds:unit rdf:resource="http://swe.nps.edu/bmds/agent/RCA1.htm"/>
1774:     <bmds:defends>
1775:     <rdf:Seq>
1776:      <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandSecurity.htm"/>
1777:      <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandCapitalBldg.htm"/>
1778:      <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandPowerPlant.htm"/>
1779:      <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandAirPort.htm"/>
1780:      <rdf:_5 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandTransport.htm"/>
1781:      <rdf:_6 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandMinOfDef.htm"/>
1782:      <rdf:_7 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandCommCtr.htm"/>
1783:      <rdf:_8 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandWtrTreatFac.htm"/>
1784:      <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/PDAL/ELandStadium.htm"/>
1785:    </rdf:Seq>
1786:   </bmds:defends>
1787:
1788:   <bmds:unit rdf:resource="http://swe.nps.edu/bmds/agent/RCA3.htm"/>
1789:     <bmds:defends>
1790:     <rdf:Seq>
1791:      <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandSecurity.htm"/>
1792:      <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandCapBuilding.htm"/>
1793:      <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandPowerPlant.htm"/>
1794:      <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandAirPort.htm"/>
1795:      <rdf:_5 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandTransport.htm"/>
1796:      <rdf:_6 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandMinOfDef.htm"/>
1797:      <rdf:_7 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandCommunicationsCenter.htm"/>
```

```
1798:      <rdf:_8 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandWtrTreatFac.htm"/>
1799:      <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/PDAL/WLandStadium.htm"/>
1800:    </rdf:Seq>
1801:   </bmds:defends>
1802:
1803: </bmds:taskToSub>
1804:
1805:
1806: </bmds:Resource>
1807:
1808:
1809: <!--END PARAGRAPH 3 EXECUTION-->
```

Listing 16. Tasks to subordinates

The execution paragraph has been broken down and described for use in the scenario described above.

6. OPORD Service Support

Paragraph 4 of the OPORD, Service Support, describes the service support areas as needed. The Service support paragraph starts at line 1817 in Listing 17. We removed the bio, relevancy, authority, and presentation blocks to save room. The internal resources for the service support paragraph are shown from lines 1871 through 1874 and consist of the material service and supplies document, health service and support document, and the personnel service support document. Each of these documents are described further; line 1881 starts the definition of the material service support section and its contents are captured at line 1934. Line 1394 starts the health service support section and line 1994 contains the content. Finally, the personnel service support section starts at line 2000 and the content at line 2053.

```
1812: <!--PARAGRAPH 4 SERVICE SUPPORT-->
1813:
1814:
1815:
1816: <!--OPORD SERVICE SUPPORT-->
1817:  <bmds:Resource rdf:about="servicespt.htm">
1818:
1819:   <!--Resource biographical information-->
1820:   <bmds:bio rdf:parseType="Resource">
.
```

```
.
.
1869:
1870:
1871: <!--Resources internal to opord EXECUTION-->
1872:   <bmds:related rdf:resource="materialServSpt.htm" />
1873:   <bmds:related rdf:resource="healthServSpt.htm" />
1874:   <bmds:related rdf:resource="personnelServSpt.htm" />
1875:
1876: </bmds:Resource>
1877:
1878:
1879:
1880: <!--OPORD MATERIAL SERVICE SUPPORT-->
1881:  <bmds:Resource rdf:about="materialServSpt.htm">
1882:
1883:   <!--Resource biographical information-->
1884:   <bmds:bio rdf:parseType="Resource">
1885:       <dc:title>MATERIAL SERVICE SUPPORT</dc:title>
1886:       <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
1887:       <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
1888:       <dc:author>COL Smith</dc:author>
1889:       <bmds:issueHq>STRATCOM</bmds:issueHq>
1890:       <bmds:classification>Unclassified</bmds:classification>
1891:       <bmds:placeIssued>
1892:         <geo:Point>
1893:           <geo:lat>20.20</geo:lat>
1894:           <geo:long>-90.80</geo:long>
1895:         </geo:Point>
1896:       </bmds:placeIssued>
1897:       <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
1898:       <bmds:orderNum>OPORD NUM</bmds:orderNum>
1899:       <bmds:codeName>BUTKUS</bmds:codeName>
1900:       <bmds:cdrLname>COOK</bmds:cdrLname>
1901:       <bmds:cdrRank>GEN</bmds:cdrRank>
1902:   </bmds:bio>
1903:
.
.
.
1933:
1934: <bmds:matServSpt>There will be Material services and support</bmds:matServSpt>
1935:
1936: </bmds:Resource>
1937:
1938:
1939:
1940: <!--OPORD HEALTH SERVICE SUPPORT-->
1941:  <bmds:Resource rdf:about="healthServSpt.htm">
1942:
1943:   <!--Resource biographical information-->
1944:   <bmds:bio rdf:parseType="Resource">
```

```
1945: <dc:title>HEALTH SERVICE SUPPORT</dc:title>
1946:    <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
1947:       <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
1948:       <dc:author>COL Smith</dc:author>
1949:       <bmds:issueHq>STRATCOM</bmds:issueHq>
1950:       <bmds:classification>Unclassified</bmds:classification>
1951:       <bmds:placeIssued>
1952:         <geo:Point>
1953:           <geo:lat>20.20</geo:lat>
1954:           <geo:long>-90.80</geo:long>
1955:         </geo:Point>
1956:       </bmds:placeIssued>
1957:       <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
.
.
.
1999:<!--OPORDPERSONNEL SERVICE SUPPORT-->
2000:<bmds:Resource rdf:about="personnelServSpt.htm">
2001:
2002:   <!--Resource biographical information-->
2003:   <bmds:bio rdf:parseType="Resource">
2004:       <dc:title>PERSONNEL SERVICE SUPPORT</dc:title>
2005:       <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
2006:       <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
2007:       <dc:author>COL Smith</dc:author>
2008:       <bmds:issueHq>STRATCOM</bmds:issueHq>
2009:       <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
2050:       </bmds:requires>
2051:   </bmds:presentation>
2052:
2053: <bmds:personServSpt>Personnel services available</bmds:personServSpt>
2054:
2055: </bmds:Resource>
2056:
2057: <!--END PARAGRAPH 4 SERVICE SUPPORT-->
```

Listing 17. Service Support

7. OPORD Command and Signal

The final paragraph of the operations order is broken down into two sub-paragraphs, Command and Signal. The command sub-paragraph contains the location and alternate location of the command post. The signal sub-paragraph lists any additional instructions not specified in Standard Operating Procedures (SOP) in addition to required reports and formats, the times the reports are to be submitted, and the chain of command. In this order we show the chain of command in lines 2117-2131.

```
2059: <!--PARAGRAPH 5 COMMAND AND SIGNAL-->
2060:
2061:
2062: <!--OPORD COMMAND AND SIGNAL-->
2063:  <bmds:Resource rdf:about="cmdSig.htm">
2064:
2065:   <!--Resource biographical information-->
2066:   <bmds:bio rdf:parseType="Resource">
2067:      <dc:title>COMMAND AND SIGNAL</dc:title>
2068:      <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
2069:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
2070:      <dc:author>COL Smith</dc:author>
2071:      <bmds:issueHq>STRATCOM</bmds:issueHq>
2072:      <bmds:classification>Unclassified</bmds:classification>
2073:      <bmds:placeIssued>
2074:        <geo:Point>
2075:          <geo:lat>20.20</geo:lat>
2076:          <geo:long>-90.80</geo:long>
2077:        </geo:Point>
2078:      </bmds:placeIssued>
2079:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
2080:      <bmds:orderNum>OPORD NUM</bmds:orderNum>
2081:      <bmds:codeName>BUTKUS</bmds:codeName>
2082:      <bmds:cdrLname>COOK</bmds:cdrLname>
2083:      <bmds:cdrRank>GEN</bmds:cdrRank>
2084:   </bmds:bio>
2085:
2086:   <!--Resource's Relevancy at time RDF/XML document was built-->
2087:   <bmds:relevancy rdf:parseType="Resource">
2088:     <bmds:currentStatus>Active</bmds:currentStatus>
2089:     <dcterms:valid >2008-01-29T00:00:00-19:00</dcterms:valid>
2090:     <dcterms:references rdf:resource="http://swe.nps.edu/bmds/oplan/lambert.htm" />
2091:     <dcterms:isReferencedBy> </dcterms:isReferencedBy>
2092:   </bmds:relevancy>
2093:
2094: <!--Resource's Authenticator-->
2095:   <bmds:authenticator rdf:parseType="Resource">
2096:     <bmds:authenName>PULFORD</bmds:authenName>
2097:     <bmds:authenPos>J3</bmds:authenPos>
2098:   </bmds:authenticator>
2099:
2100: <!--Resource's presentation/consumption information about resource-->
2101:   <bmds:presentation rdf:parseType="Resource">
2102:      <dc:format>text/html</dc:format>
2103:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
2104:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
2105:      <dcterms:requires>HTML User agent</dcterms:requires>
2106:      <bmds:requires rdf:parseType="Resource">
2107:        <bmds:type>stylesheet</bmds:type>
2108:        <rdf:value>http://swe.nps.edu/opord.css</rdf:value>
2109:      </bmds:requires>
```

```
2110:      <bmds:requires rdf:parseType="Resource">
2111:        <bmds:type>logo</bmds:type>
2112:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
2113:      </bmds:requires>
2114:   </bmds:presentation>
2115:
2116:
2117:  <bmds:chainOfCmd>
2118:     <rdf:Seq>
2119:      <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/agent/sca1.htm"/>
2120:      <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/agent/rca2.htm"/>
2121:      <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/agent/rca1.htm"/>
2122:      <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/agent/rca3.htm"/>
2123:      <rdf:_5 rdf:resource="http://swe.nps.edu/bmds/agent/tca21.htm"/>
2124:      <rdf:_6 rdf:resource="http://swe.nps.edu/bmds/agent/tca22.htm"/>
2125:      <rdf:_7 rdf:resource="http://swe.nps.edu/bmds/agent/tca11.htm"/>
2126:      <rdf:_8 rdf:resource="http://swe.nps.edu/bmds/agent/tca12.htm"/>
2127:      <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/agent/tca13.htm"/>
2128:      <rdf:_8 rdf:resource="http://swe.nps.edu/bmds/agent/tca31.htm"/>
2129:      <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/agent/tca32.htm"/>
2130:    </rdf:Seq>
2131:  </bmds:chainOfCmd>
2132:
2133: <bmds:signal>Per SOI</bmds:signal>
2134: </bmds:Resource>
2135: <!--END PARAGRAPH 5 COMMAND AND SIGNAL-->
2136:
2137:
2138: <!--END RELATED RESOURCES-->
2139:
2140:
2141:
2142: </rdf:RDF>
```

Listing 18. Command and Signal

We have just shown a complete RDF/XML for the top-level command within our BMDS scenario. This complete order would then be sent to all of its subordinate commands that in this example consist of three Regional Command Agents (RCA), those being RCA1, RCA2, and RCA3. Next we show the order that is created and issued by RCA2 upon receipt of the above SCA1 OPORD.

8. Regional Command Agent Operations Order

The regional command agents are the next level of command hierarchy in our scenario. They receive the OPORD from SCA1, process it, write a new order and issue it

to their subordinates. We show in the next several sections RCA2's OPORD. We strip out all of the bio, relevancy, authenticator, and presentation code as we have shown it on several previous occasions. Also, as the order format does not change, we show only the significant differences between the SCA1 order and the RCA2 order and those paragraphs or sub-paragraphs that are taken part and parcel from SCA1.

```
<?xml version="1.0"?>
2: <rdf:RDF xml:lang="en"
3:     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:     xmlns:bmds="http://swe.nps.edu/bmds/elements/1.0/"
5:     xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
6:     xmlns:dcterms="http://purl.org/dc/terms"
7:     xmlns:dc="http://purl.org/dc/elements/1.1/"
8:     xml:base="http://swe.nps.edu/bmds/opords/rca2/">
9:
10: <bmds:Resource rdf:about="rca220080129.htm">
```

Listing 19. RCA2 OPORD

The first important change we must note is that the namespace base for this OPORD has changed. In Listing 19 at line 8 the base for this OPORD is listed as http://swe.nps.edu/bmds/opords/rca2/. This allows us to use the short names such as "rca220080129.htm". When parsed, the addition of the base is automatically inserted to creat the entire namespace http://swe.nps.edu/bmds/opords/rca2/rca220080129.htm. This prevents namespace collisions with the SCA1 OPORD.

Next, in Listing 20 we show the header information that is attached to each resource that is described for each resource in the document RCA2 OPORD. This information, just as in SCA1 OPORD, consists of the biographical, relevancy, authenticator, and presentation meta data. The difference between the two is the information the meta-data describes. A few examples are as follows. The dc:title in Listing 20 line 14 is 'opordrca220080202, at line 15 dc:dateCreated is "2008-02-02T00:00:00-09:00", at line 17 dc:author is "COL BARD", and at line 29 the bmds:cdrLname is "Hart". The presentation meta-data and encapsulated data from line 49 through 63 remains identical to the SCA1 OPORD as the means of presenting and consuming the information is the same. Lines 68 through lines 73 of the RCA2 OPORD

in Listing 20 are identical to lines 67-71 of SCA1 OPORD in Listing 5; they define the five paragraphs of there respective OPORDs, it is the namespaces that differentiates between the two.

In the next listing we look at the Task Organization annex, one of the internal related resources identified in Listing 20 at line 74.

```
<bmds:Resource rdf:about="rca220080129.htm">
11:
12:   <!--Resource biographical information-->
13:   <bmds:bio rdf:parseType="Resource">
14:      <dc:title>opord rca220080202</dc:title>
15:      <dc:dateCreated>2008-02-02T00:00:00-09:00</dc:dateCreated>
16:      <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
17:      <dc:author>COL Bard</dc:author>
18:      <bmds:issueHq>JTF 1</bmds:issueHq>
19:      <bmds:classification>Unclassified</bmds:classification>
20:      <bmds:placeIssued>
21:        <geo:Point>
22:          <geo:lat>32.30</geo:lat>
23:          <geo:long>45.54</geo:long>
24:        </geo:Point>
25:      </bmds:placeIssued>
26:      <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
27:      <bmds:orderNum>01Thomas</bmds:orderNum>
28:      <bmds:codeName>Thomas</bmds:codeName>
29:      <bmds:cdrLname>Hart</bmds:cdrLname>
30:      <bmds:cdrRank>GEN</bmds:cdrRank>
31:   </bmds:bio>
32:
33:   <!--Resource's Relevancy at time RDF/XML document was built-->
34:   <bmds:relevancy rdf:parseType="Resource">
35:     <bmds:currentStatus>Active</bmds:currentStatus>
36:     <dcterms:valid >2008-02-02T00:00:00-09:00</dcterms:valid>
37:     <dcterms:references rdf:resource="http://swe.nps.edu/bmds/opord/rca2/rca220080202.htm" />
40:     <dcterms:isReferencedBy> </dcterms:isReferencedBy>
41:   </bmds:relevancy>
42:
43: <!--Resource's Authenticator-->
44:   <bmds:authenticator rdf:parseType="Resource">
45:     <bmds:authenName>Jones</bmds:authenName>
46:     <bmds:authenPos>J3</bmds:authenPos>
47:   </bmds:authenticator>
48:
49: <!--Resource's presentation/consumption information about resource-->
50:   <bmds:presentation rdf:parseType="Resource">
51:      <dc:format>text/html</dc:format>
52:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
```

```
53:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
54:      <dcterms:requires>HTML User agent</dcterms:requires>
55:      <bmds:requires rdf:parseType="Resource">
56:        <bmds:type>stylesheet</bmds:type>
57:        <rdf:value>http://swe.nps.edu/opord.css</rdf:value>
58:      </bmds:requires>
59:      <bmds:requires rdf:parseType="Resource">
60:        <bmds:type>logo</bmds:type>
61:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
62:      </bmds:requires>
63:   </bmds:presentation>
64:
65:
66:
67:
68: <!--Resources internal to opord RCA220080202-->
69:   <bmds:related rdf:resource="situation.htm" />
70:   <bmds:related rdf:resource="mission.htm" />
71:   <bmds:related rdf:resource="execution.htm" />
72:   <bmds:related rdf:resource="serviceSpt.htm" />
73:   <bmds:related rdf:resource="cmdSig.htm" />
74:   <bmds:related  rdf:resource="http://swe.nps.edu/bmds/opord/rca2/annexs/annexATaskO.htm" />
```

Listing 20. RCA2 OPORD Header

In Listing 21 we show the task organization of RCA2. Line 12 identifies the resource we are referring to, Annex A Task Organization, and from lines 15 through 18 we show the roles and relationship to RCA2. In line 15 RCA2 is identified as the commander using the bmds:cdr tag. In lines 16 and 17 we identify TCA 21 and TCA 22 as subordinates to RCA2 using the bmds:subordinate tag. In line 18 FBX21 is identified as a sensor that belongs to RCA2. This is similar to how SCA1's task organization would look, identifying RCA1, RCA2, and RCA3 as subordinates to SCA1.

```
1: <?xml version="1.0"?>
2: <rdf:RDF xml:lang="en"
3:     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:     xmlns:bmds="http://swe.nps.edu/bmds/elements/1.0/"
5:     xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
6:     xmlns:dcterms="http://purl.org/dc/terms"
7:     xmlns:dc="http://purl.org/dc/elements/1.1/"
8:     xml:base="http://swe.nps.edu/bmds/opord/rca2/annexs/">
9:
10: <!-- TaskO RCA2 -->
11:
12: <rdf:Description rdf:about="annexATaskO.htm">
13:
15: <bmds:cdr rdf:resource="http://swe.nps.edu/bmds/agent/RCA2"/>
16: <bmds:subord rdf:resource="http://swe.nps.edu/bmds/agent/TCA21"/>
17: <bmds:subord rdf:resource="http://swe.nps.edu/bmds/agent/TCA22"/>
18: <bmds:sensor rdf:resource="http://swe.nps.edu/bmds/sensor/FBX21"/>
19:
21: </rdf:Description>
24: </rdf:RDF>
25:
```

Listing 21. Task Organization

We continue our look at RCA2's main OPORD below in Listing 22 looking at lines 656 and 657. These lines identify the friendly forces sketch from SCA1's OPORD and the related sketch; this is beneficial as we do not need to produce anything further because here we reuse the information from SCA1 OPORD.

We do much of the same with the Mission paragraph shown from lines 1104 through 1383. The who, what, where, when, and why content changes only slightly. The who in line 1106 Listing 22 refers now to RCA2, the what in line 1107 refers to RCA2's PDAL, the where in line 1108 again refers to RCA2's PDAL, the when is no change from SCA1 OPORD as the time to establish the defense never changes for all OPORDs, and the why at line 1110 is slightly modified by the J3 staff of RCA2. From lines 1175 through 1198 the order shows the assets in priority order that belong to RCA2; they originate from the SCA1 order. Each of the assets is also identified as a resource and have the same unique identifier whether they are referenced by the SCA, RCA, or any other agent or resource. Lines 1292-1383 show the properties of the individual defended assets as discussed earlier.

<!--Resource's presentation/consumption information about resource-->
640:
.
.
.
654:
655: <!--Resources internal to FRIENDLY FORCES-->
656: <bmds:related
rdf:resource="rdf:resource="http://swe.nps.edu/bmds/opord/sca120080129.htm/higherHq.htm" />
657: <bmds:related
rdf:resource="http://swe.nps.edu/bmds/opord/sca120080129.htm/friendlyForcesSketch.htm" />
658:
659: </bmds:Resource>

1104: <!--OPORD MISSION STATEMENT-->
1105: <bmds:mission rdf:parseType="Resource">
1106: <bmds:who rdf:resource="rca2.htm" />
1107: <bmds:what rdf:resource="http://swe.nps.edu/bmds/ujtl/st6-1-5.htm" />
1108: <bmds:where rdf:resource="rca2PDAL.HTML" />
1109: <bmds:when>2008-02-23T00:00:00-04:00</bmds:when>
1110: <bmds:why>Preservation of peace in the local region</bmds:why>
1111: </bmds:mission>
1115: <!--Resources internal to mission-->
1116: <bmds:related rdf:resource="rca2PDAL.htm" />
1117:
1118: </bmds:Resource>
1119:
1121: <bmds:Resource rdf:about="rca2PDAL.htm">
1122:
1123: <!--Resource biographical information-->
1124: <bmds:bio rdf:parseType="Resource">
1125: <dc:title>RCA2 PDAL</dc:title>
1126: <dc:dateCreated>2008-02-02T00:00:00-09:00</dc:dateCreated>
1142: </bmds:bio>
.
.
.
1174:
1175: <!--OPORD PDAL-->
1176: <bmds:pdal>
1177: <rdf:Seq>
1178: <rdf:_1rdf:resource="http://swe.nps.edu/bmds/PDAL/Island_X.htm"/>
1179: <rdf:_2rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm"/>
1182:<rdf:_3rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm"/>
1185:<rdf:_4rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXAirPort.htm"/>
1186: <rdf:_5rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXShippingPort.htm"/>
1189: <rdf:_6rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYMinOfDef.htm"/>
1192: <rdf:_7rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCommCntr.htm"/>
1195: <rdf:_8rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYWtrTreatFac.htm"/>
1198: <rdf:_9 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCapitalCity.htm"/>
1201:</rdf:Seq>
1202: </bmds:pdal>

```
1205: <!--Resources internal to mission-->
1206:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/ujtl/st6-1-5.htm" />
1207:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/PDAL/Island_X.htm"/>.
.
.
1226:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYWtrTreatFac.htm"/>
1229:   <bmds:related  rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCapitalCity.htm"/>
1232:   <bmds:related
1289: </bmds:Resource>
1290:
1291:
1292: <rdf:Description rdf:about="http://swe.nps.edu/bmds/PDAL/Island_X.htm">
1293:    <bmds:location>
1294:     <geo:Point>
1295:       <geo:lat>62.20</geo:lat>
1296:         <geo:long>80.80</geo:long>
1297:       </geo:Point>
1298:    </bmds:location>
1299:   <bmds:assetType>land mass</bmds:assetType>
1300: </rdf:Description>
1301:
1302: <rdf:Description rdf:about="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm">
1305:    <bmds:location>
1306:     <geo:Point>
1307:       <geo:lat>58.20</geo:lat>
1308:         <geo:long>78.80</geo:long>
1309:       </geo:Point>
1310:    </bmds:location>
1311:  <bmds:assetType>Building</bmds:assetType>
1312: </rdf:Description>
1313: <rdf:Description rdf:about="http://swe.nps.edu/bmds/PDAL/MainLand_Y/CapitalCity">
1375:   <bmds:location>
1376:     <geo:Point>
1377:       <geo:lat>20.20</geo:lat>
1378:         <geo:long>-90.80</geo:long>
1379:       </geo:Point>
1380:    </bmds:location>
1381:   <bmds:assetType>Capital City</bmds:assetType>
1382:   <bmds:population>1,241,045</bmds:population>
1383: </rdf:Description>
```

Listing 22. RCA2 OPORD

The last piece of the mission statement is the associated MOEs. They do not change from the SCA1 order and are listed in Listing 12, lines 1390-1410. and their associated definitions in lines 1416-1503.

Next in Listing 23 we show the relevant piece of the execution paragraph; the task to subordinate units. From Line 1999 through 2025 the PDAL is further broken down and the defended assets are assigned to TCA22 and TCA21 respectively.

```
1997: <bmds:taskToSub rdf:parseType="Resource">
1998:
1999:   <bmds:unit rdf:resource="http://swe.nps.edu/bmds/agent/TCA22.htm"/>
2000:     <bmds:defends>
2001:     <rdf:Seq>
2002:      <rdf:_1rdf:resource="http://swe.nps.edu/bmds/PDAL/Island_X.htm"/>
2003: <rdf:_2:rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm"/>
2006:<rdf:_3rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm"/>
2009:<rdf:_4rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXAirPort.htm"/>
2010: <rdf:_5rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXShippingPort.htm"/>
2013:</rdf:Seq>
2014: </bmds:defends>
2015: <bmds:unitrdf:resource="http://swe.nps.edu/bmds/agent/TCA21.htm"/>
2016:    <bmds:defends>
2017 :     <rdf:Seq>
2018:<rdf:_6 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYMinOfDef.htm"/>
2020:<rdf:_7rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCommCntr.htm"/>
2022: <rdf:_8rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYWtrTreatFac.htm"/>
2024:<rdf:_9rdf:resource=”http://swe.nps.edu/bmds/PDAL/MnLandYCapitalCity.htm"/>
2026:    </rdf:Seq>
2027:   </bmds:defends>
2028: </bmds:taskToSub>
```

Listing 23. RCA2 OPORD Execution Paragraph

The final piece of the RCA2 OPORD we show below in Listing 24 is the chain of command. It simply shows that the succession of command for the unit; RCA2 followed by TCA21 and then TCA22.

```
2405:  <bmds:chainOfCmd>
2406: <rdf:Seq>
2408:   <rdf:_1rdf: resource="http://swe.nps.edu/bmds/agent/rca2.htm"/>
2411:   <rdf:_2rdf: resource="http://swe.nps.edu/bmds/agent/tca21.htm"/>
2412:   <rdf:_3rdf: resource="http://swe.nps.edu/bmds/agent/tca22.htm"/>
2418:    </rdf:Seq>
2419:  </bmds:chainOfCmd>
```

Listing 24. RCA2 Chain of Command

Thus far we have described two operations orders: one developed by SCA1 and the other developed by RCA2. We have shown that the one developed by RCA2 is a refinement of the one developed by SCA1. In the next portion of this section we will show how upon receipt of the RCA2 order its subordinates (i.e., TCA21 and TCA22) further refine the RCA2 order and develop their own order, which they subsequently issue directly to the battle managers of the weapons systems and sensors. As the two orders produced by TCA21 and TCA22 will be similar we will show only portions of TCA21 to avoid repetition.

9. Tactical Command Agent Operations Order

The definition for the TCA21 OPORD is listed in Listing 25 below. The OPORD format is identical to the two previous higher level orders and the content is unit appropriate. In a couple of follow on tables, we show the significant content differences between the received order and the order issued to the subordinates. We point out that the recipients of the order in Listing 25 will be the associated weapons and sensor systems.

```
1: <?xml version="1.0"?>
2: <rdf:RDF xml:lang="en"
3:     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:     xmlns:bmds="http://swe.nps.edu/bmds/elements/1.0/"
5:     xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
6:     xmlns:dcterms="http://purl.org/dc/terms"
7:     xmlns:dc="http://purl.org/dc/elements/1.1/"
8:     xml:base="http://swe.nps.edu/bmds/opords/tca21/">
9:
10: <bmds:Resource rdf:about="tca210080129.htm">
11:
12:   <!--Resource biographical information-->
13:   <bmds:bio rdf:parseType="Resource">
14:       <dc:title>opord tca210080129</dc:title>
15:       <dc:dateCreated>2008-02-02T00:00:00-20:30</dc:dateCreated>
16:       <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
17:       <dc:author>COL Mann</dc:author>
18:       <bmds:issueHq>TCA 21</bmds:issueHq>
19:       <bmds:classification>Unclassified</bmds:classification>
20:       <bmds:placeIssued>
21:         <geo:Point>
22:           <geo:lat>60.50</geo:lat>
23:           <geo:long>60.06</geo:long>
```

```
24:        </geo:Point>
25:      </bmds:placeIssued>
26:      <bmds:msgRefNum>456789</bmds:msgRefNum>
27:      <bmds:orderNum>08</bmds:orderNum>
28:      <bmds:codeName>Hansen</bmds:codeName>
29:      <bmds:cdrLname>Gilbert</bmds:cdrLname>
30:      <bmds:cdrRank>GEN</bmds:cdrRank>
31:   </bmds:bio>
32:
33:   <!--Resource's Relevancy at time RDF/XML document was built-->
34:   <bmds:relevancy rdf:parseType="Resource">
35:     <bmds:currentStatus>Active</bmds:currentStatus>
36:     <dcterms:valid >2008-02-02T00:00:00-20:30</dcterms:valid>
37:     <dcterms:references rdf:resource="http://swe.nps.edu/bmds/opord/rca2/rca220080202.htm" />
38:     <dcterms:isReferencedBy> </dcterms:isReferencedBy>
39:   </bmds:relevancy>
40:
41: <!--Resource's Authenticator-->
42:   <bmds:authenticator rdf:parseType="Resource">
43:     <bmds:authenName>James</bmds:authenName>
44:     <bmds:authenPos>Asst J3</bmds:authenPos>
45:   </bmds:authenticator>
46:
47: <!--Resource's presentation/consumption information about resource-->
48:   <bmds:presentation rdf:parseType="Resource">
49:      <dc:format>text/html</dc:format>
50:      <dcterms:conformsTo>XHTML 1.0 Strict</dcterms:conformsTo>
51:      <dcterms:conformsTo>CSS Validation</dcterms:conformsTo>
52:      <dcterms:requires>HTML User agent</dcterms:requires>
53:      <bmds:requires rdf:parseType="Resource">
54:        <bmds:type>stylesheet</bmds:type>
55:        <rdf:value>http://swe.nps.edu/opord.css</rdf:value>
56:      </bmds:requires>
57:      <bmds:requires rdf:parseType="Resource">
58:        <bmds:type>logo</bmds:type>
59:        <rdf:value>http://swe.nps.edu/logo/nps.jpg</rdf:value>
60:      </bmds:requires>
61:   </bmds:presentation>
62:
63:
64:
65:
66: <!--Resources internal to opord tca210080129-->
67:   <bmds:related rdf:resource="situation.htm"/>
68:   <bmds:related rdf:resource="mission.htm"/>
69:   <bmds:related rdf:resource="execution.htm"/>
70:   <bmds:related rdf:resource="serviceSpt.htm"/>
71:   <bmds:related rdf:resource="cmdSig.htm"/>
72:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexATaskO.htm" />
73:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexBIntel.htm" />
74:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexCOps.htm" />
75:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexDLog.htm" />
```

```
76:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexEPers.htm" />
77:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexFPubAffair.htm" />
78:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexGCivAffair.htm" />
79:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexHMetoc.htm" />
80:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexJCmdRel.htm" />
81:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexKC3.htm" />
82:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexLEnviron.htm" />
83:   <bmds:related
rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexMMapChrtGeo.htm" />
84:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexNSpaceOps.htm" />
85:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexPHostNatSpt.htm"
/>
86:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexQMedServ.htm" />
87:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexSSpecTechOps.htm"
/>
88:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexTConsMgt.htm" />
89:   <bmds:related
rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexVIntAgentCoord.htm" />
90:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexXExeChkList.htm"
/>
91:   <bmds:related rdf:resource="http://swe.nps.edu/bmds/opord/tca21/annexs/annexZDistro.htm" />
92:
93:
94: </bmds:Resource>
```

Listing 25. TCA 21 OPORD

One of the main differences in the RCA2 order and the TCA21 order is that the PDAL is now refined to a point where weapon systems are assigned the responsibility of defending a particular asset as shown below in Listing 26. Line 1575 identifies the unit, GMD211, as the weapon system responsible for defending, in priority order, line 1578: Island X's capitol building, line 1579: Island X's power plant, line 1580: the main land Y's Ministry of Defense, and line 1581: mainland Y's capital city. Next, line 1582 identifies THAAD211 as the weapon system responsible for defending, in priority order, line 1588: island X's power plant, line 1589: Island X's shipping port, line 1590: main land Y's communications center, and line 1591 mainland Y's water treatment facility. The final weapon system, ABL211, is shown at line 1595 and defends Island X's power plant, mainland Y's communications center, and mainland Y's water treatment facility at lines 1598, 1599, and 1600, respectively.

```
1519: <!--OPORD EXECUTION, TASKS TO SUBORDINATES -->
1520:  <bmds:Resource rdf:about="taskToSubs.htm">
1521:
1572:
1573: <bmds:taskToSub rdf:parseType="Resource">
1574:
1575:   <bmds:unit rdf:resource="http://swe.nps.edu/bmds/weapon/GMD211.htm"/>
1576:    <bmds:defends>
1577:    <rdf:Seq>
1578:     <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXCap_Building.htm"/>
1579:     <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm"/>
1580:     <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYMinOfDef.htm"/>
1581:     <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCapitalCity.htm"/>
1582:    </rdf:Seq>
1583:    </bmds:defends>
1584:
1585:   <bmds:unit rdf:resource="http://swe.nps.edu/bmds/weapon/THAAD211.htm"/>
1586:    <bmds:defends>
1587:    <rdf:Seq>
1588:     <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm"/>
1589:     <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXShippingPort.htm"/>
1590:     <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCommCntr.htm"/>
1591:     <rdf:_4 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYWtrTreatFac.htm"/>
1592:   </rdf:Seq>
1593:  </bmds:defends>
1594:
1595:  <bmds:unit rdf:resource="http://swe.nps.edu/bmds/weapon/ABL211.htm"/>
1596:    <bmds:defends>
1597:    <rdf:Seq>
1598:     <rdf:_1 rdf:resource="http://swe.nps.edu/bmds/PDAL/IslXPowerPlant.htm"/>
1599:     <rdf:_2 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYCommCntr.htm"/>
1600:     <rdf:_3 rdf:resource="http://swe.nps.edu/bmds/PDAL/MnLandYWtrTreatFac.htm"/>
1601:   </rdf:Seq>
1602:  </bmds:defends>
1603:
1604: </bmds:taskToSub>
1605:
1606: </bmds:Resource>
```

Listing 26. TCA21 Task to Subordinates

10. Summary

In Section 4 of this chapter, we showed the development of machine and human readable operations orders using RDF/XML for three levels of command and for weapons and sensor systems. We showed the similarities of the orders and how many sections were reused in subsequent lower level orders.

In Chapter 4 of the dissertation we show how the orders and pieces of the orders can be passed between battle managers (e.g. SCA's, RCA's, and TCA's), weapons, and sensors in order to establish the battle field environment prior to hostilities and then how the battle managers can then execute the battle based on the machine-processable OPORDs.

IV. SERVICE ORIENTED BALLISTIC MISSILE DEFENSE COMMAND CONTROL AND BATTLE MANAGEMENT

A. INTRODUCTION

C4ISR applications require making decisions based on situational awareness created by fusing sensory information collected from independently maintained sources. Having a C2 structure that respects the autonomy of basic services facilitates the flexibility to dynamically negotiate and adjust to changes in the battle space while maintaining the both continuity of the overall operations and deployment readiness. In this chapter, we develop such a framework to thwart threats from ballistic missiles by using a three-tiered C2 structure: This structure is congruent with the U.S. DoD's objective of adopting Service oriented Architecture (SOA) in which the master orchestrator provides a service by composing the services of the autonomously functioning sub-services. We model the continuity of orchestrated operations via duty cycles, with each duty cycle reacting to environmental changes. The orchestrator provides the required QoS – which includes timeliness as one aspect [26]. As shown in this chapter, a flexible QoS-sensitive SOA suffices to specify and implement stated C4ISR requirements.

The DoD mandated the basic WS framework standards for use in Enterprise Resource Planning (ERP) software packages, but has not mandated the standards for use in the GIG as the standards fall short in meeting GIG security and authorization requirements [27]. The basic WS framework standards include what are commonly referred to as the *core* WS development standards: WSDL, SOAP, and UDDI. Although the core WS have been applied successfully by industry in business systems, as Birman et al. claim [28], these systems fall short of C4ISR needs due to the lack of support for time-critical events. Consequently, in this study we decorate BPEL-specified duty cycles with QoS, specifically timeliness attributes, MOP and MOE specifications, with the hope that a SOA satisfying the need articulated by Birman et al. can implement our design. Section B of this chapter contains the use cases for a ballistic missile defense system. Section C presents an overview of conventional C2 and possible execution using WS and specifies

a C2 family of WS using WSDL. Section D describes the OPORD, and Section E describes how BPEL can be used for process integration. Section F describes the evolution of the OPORD. Section G provides our conclusions.

B. BMD C2

The objective of the Missile Defense Agency's (MDA) Advanced Battle Manager (ABM) of the Ballistic Missile Defense System (BMDS) [9] is to provide an integrated, layered defense from ballistic missiles of all ranges in all phases of their flight. At a high level, BMDS consist of an integrated C2, Battle Management (BM), and Communications (collectively known as C2BMC), and weapons and sensors. Weapon and sensors are capable of engaging and sensing many different threat missiles through different phases of their flight: boost, mid-course and terminal. The C2 component is responsible for creating and distributing the OPORD) that essentially provides initial weapons, and sensor locations, their orientations, and their responsibilities within the plan while the BM executes the battle according to the OPORD and the responses from sensory inputs.

Wijesekera, Michael and Nerode [29] use three kinds of agents to model BMDS C2: the strategic commander agent (SCA), regional commander agent (RCA), and the tactical commander agent (TCA). Each battle manager assumes one of these roles. A hierarchical command structure in [29] consists of SCAs at the top of the C2 structure that share information horizontally between them, As shown in Figure 16, each SCA manages vertically down the chain to its assigned RCA's and with any assigned sensors and weapons in the sensors and weapons nets. Continuing down the C2 structure the individual RCA's manage and communicate with their assigned TCAs and any assigned weapons and sensors in the weapons and sensors nets. Finally, each TCA manages and communicates with its assigned weapons and sensors within the sensors and weapons nets. While information travels up and down the C2 structure, most down-flows are commands and most up-flows are status reports.

1. C2 Structure Scenario

The top of the Chain of Command has a single SCA. In our scenario, SCA1 if necessary coordinates with other SCAs on the BMDS network. SCA1 commands three regional agents: RCA1, RCA2, and RCA3. SCA1 also controls two standalone radars: RadarS1 and RadarS2. RCA1 commands three tactical agents: TCA11, TCA12, and TCA13. RCA1 also controls a standalone radar, RadarR11. RCA2 commands two tactical agents, TCA21, TCA22, in addition to controlling a standalone radar, RadarR21. RCA3 commands two tactical agents TCA31 and TCA32. At the tactical level each TCA manages some set of weapons systems and sensors. In this scenario, TCA11 manages a Ground Missile Defense System (GMD), Terminal High Altitude Area Defense (THAAD) system, and a Patriot Missile Battalion. The rest of the scenario C2 structure is shown in Figure 16.

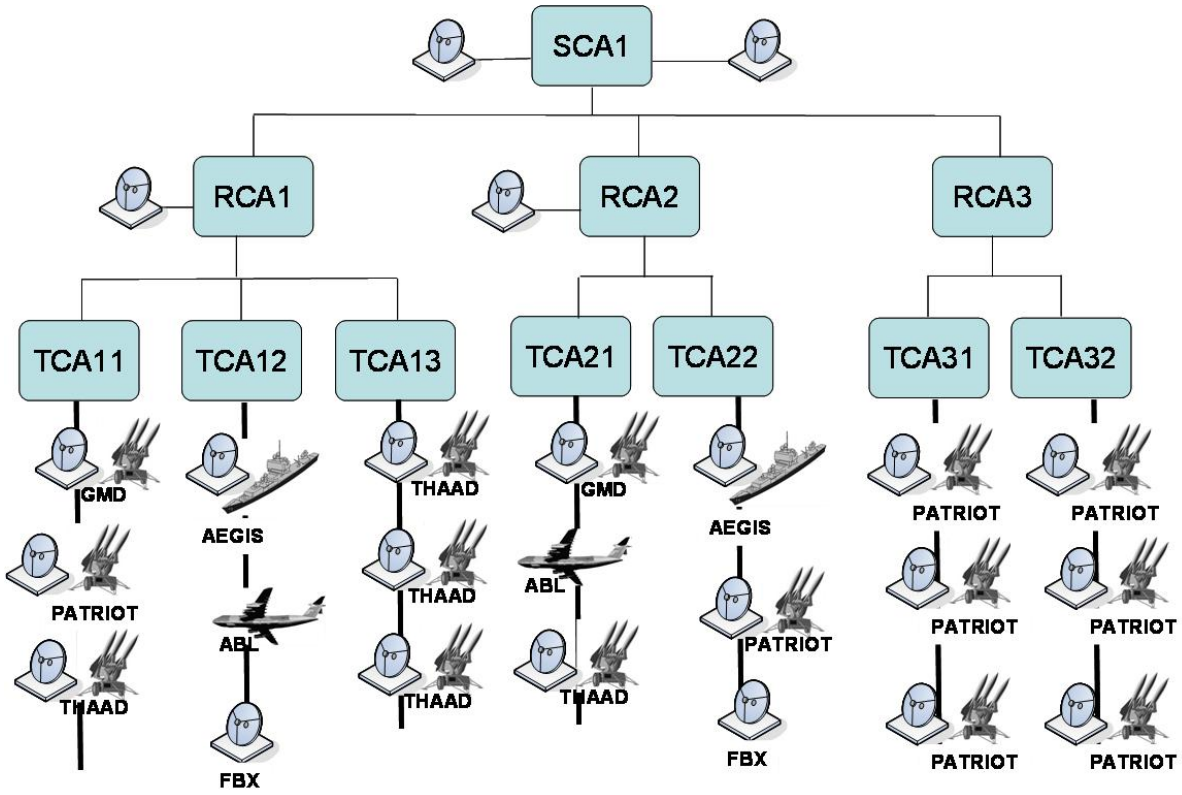

Figure 16. BMDS C2 structure

2. Assumptions

a. OPORDs have been issued by all agents and each agent has established its defenses. This implies that all weapon and sensor systems for the entire BMDS are positioned to defend against the most likely threat missile attack according to the OPORD intelligence estimate; weapon and sensor systems have specific orientations ready to launch and sensors are in surveillance mode.

b. If the enemy attacks according to the OPORD intelligence estimate, the weapon and sensor systems execute the plan autonomously with little or no interference from the command agents.

c. The the enemy does not attack according to the intelligence estimate and SCA1, RCA2 and assigned TCAs must manage the initial attack to defeat the threats.

3. Scenario Execution

Our scenario shown in Figure 17 proceeds as follows. RCA2's organic sensor and BM determine that three separate threat missiles are inbound and predicted to hit a high-priority asset on its Prioritized Defended Asset List (PDAL). According to the OPROD the terminal-defense mission for the asset being attacked is assigned to TCA22 and the midcourse defense of the asset is assigned to TCA21. However, based on the OPORD Intel annex most resources in SCA1's area of operation are oriented on the enemy's likely air avenue of attack, depicted by the large dotted arrow and labeled as such in the figure. Therefore, reorienting of resources within RCA2's area of operation is necessary to negate the threats. RCA2 concurrently sends a contact message to SCA1 requesting permission to reorient resources and engage the threats, in addition to sending a be-prepared-to-launch order to TCA21 and TCA22.

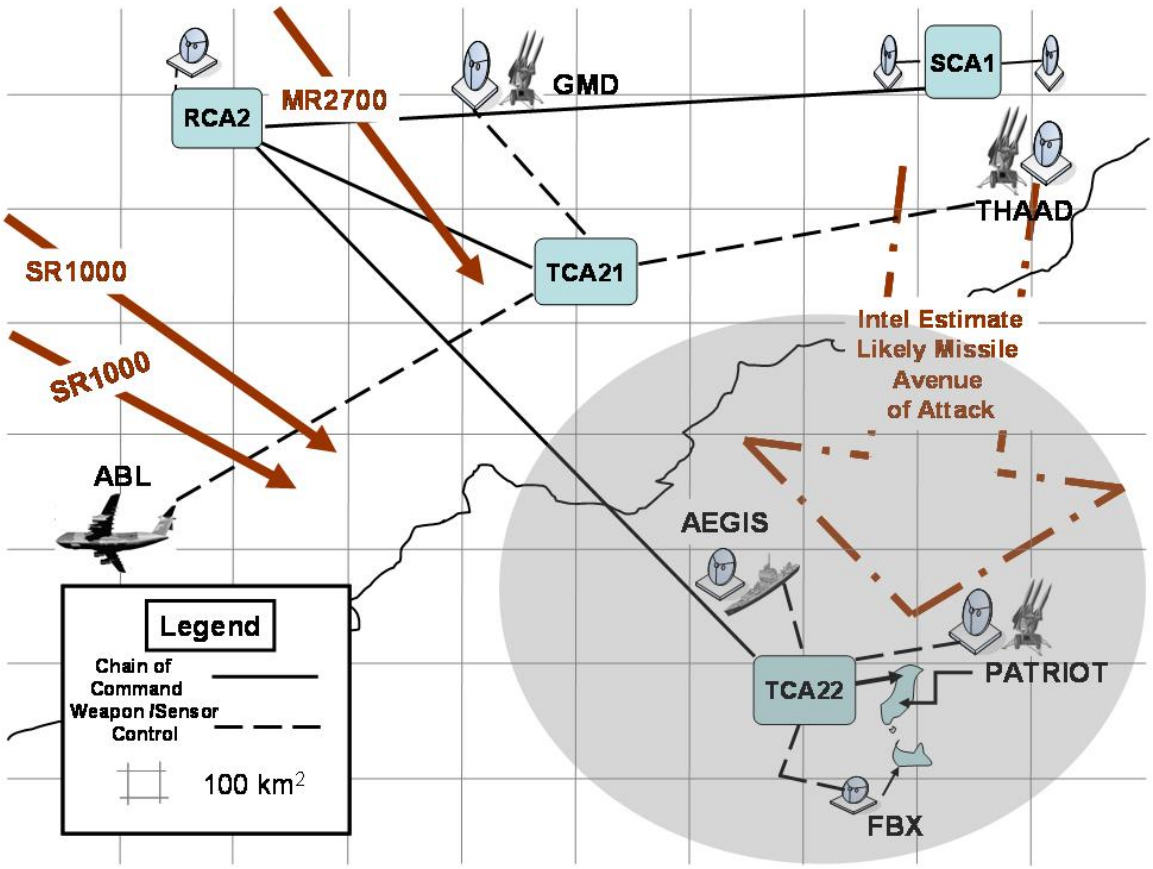

Figure 17. BMDS Scenario

The messages exchanged include the individual threat tracks and values for specific QoS attributes, MOP, and MOE necessary to ensure the threats are engaged prior to reaching their keep-out ranges. In this scenario, RCA2 has two MOEs associated with it: (i) survivability, defined as the fraction of defended assets that survive the attack and (ii) the probability that the interceptor kills the threat target given that it arrives in time. The MOP associated with RCA2 is Time-on-Target; the time remaining for any weapon system to launch an interceptor is also included giving subordinates an upper bound on the time to engage with the appropriate shot doctrine.

TCA21 receives the be-prepared order (to launch) and steps into the kill-chain cycle at the assign weapon task. Using the track information from RCA2, TCA21 determines the appropriate weapon systems with which to engage the threat missiles, builds an engagement plan, and issues a be-prepared order to the appropriate weapons

and sensors. Likewise, TCA22 receives its be-prepared order to launch, but its launch is contingent on the threat reaching the keep-out range. TCA22 also steps into the kill chain cycle at the assign weapons task and issues be-prepared missions to its associated weapons and sensors.

Upon receiving SCA1's response message to launch, RCA2 issues a message to the TCAs to execute the be-prepared missions. TCAs 21 and 22 use the MOEs and MOP from RCA2 to guide the selection of the services required to complete the weapons assignment, engagement, and assess kill tasks of the kill chain. We show this in detail in subsequent sections, but first we describe the scenario using use case notation for each of the agents: SCA1, RCA2, and TCAs 21 and 22. We now describe, in use case (a technique for describing how to achieve a goal or task) format, the kill-chain tasks performed by each agent in our scenario. Figure 18 shows the process logic a BM executes upon receiving a track list.

Use Case 1: *Detect*

Goal in Context: Identify threats from a list of reported sensor tracks

Scope & Level: A primary task of the battle Manager

Preconditions: Battle manager has been initialized

Success End Condition: Correctly identify threat object.

Failed End Condition: Fails to identify threat missile.

Primary Actor: Battle Manager

Trigger: Receive track list from a sensor

MAIN SUCCESS SCENARIO

1. Receive Track List message

2. Verifies source of message

3. Validate request parameters

4. Associates Track List

5. Correlates Track List

6. Returns a threat list

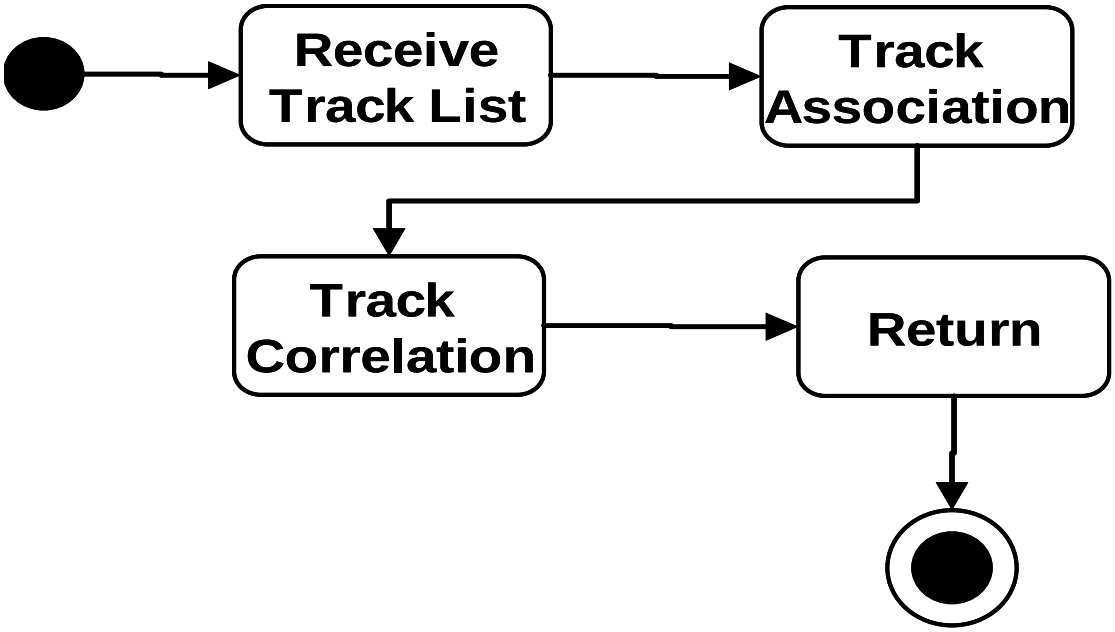

Figure 18. Detect

Figure 19 shows the execution logic of a BM upon receipt of a threat track list from a sensor in its C2 structure.

Use Case 2: *Track*

Goal in Context: Return a launch quality threat track

Scope & Level: A primary task of the battle Manager

Preconditions: Detect Task complete

Success End Condition: Produces fire quality tracks

Failed End Condition: Fails to produce fire quality track

Primary Actor: Battle Manager

Trigger: Receive a threat list

MAIN SUCCESS SCENARIO

1. Receive Threat List message

2. Verifies source of message

3. Validate request parameters

4. Fuse threat List

5. Calculate IPP for threats

6. Calculate Aim Point for threats

7. Calculate time available to kill threats

8. Get QoS, MOP, MOE requirements

9. Returns a threat list, QoS, MOP, MOE

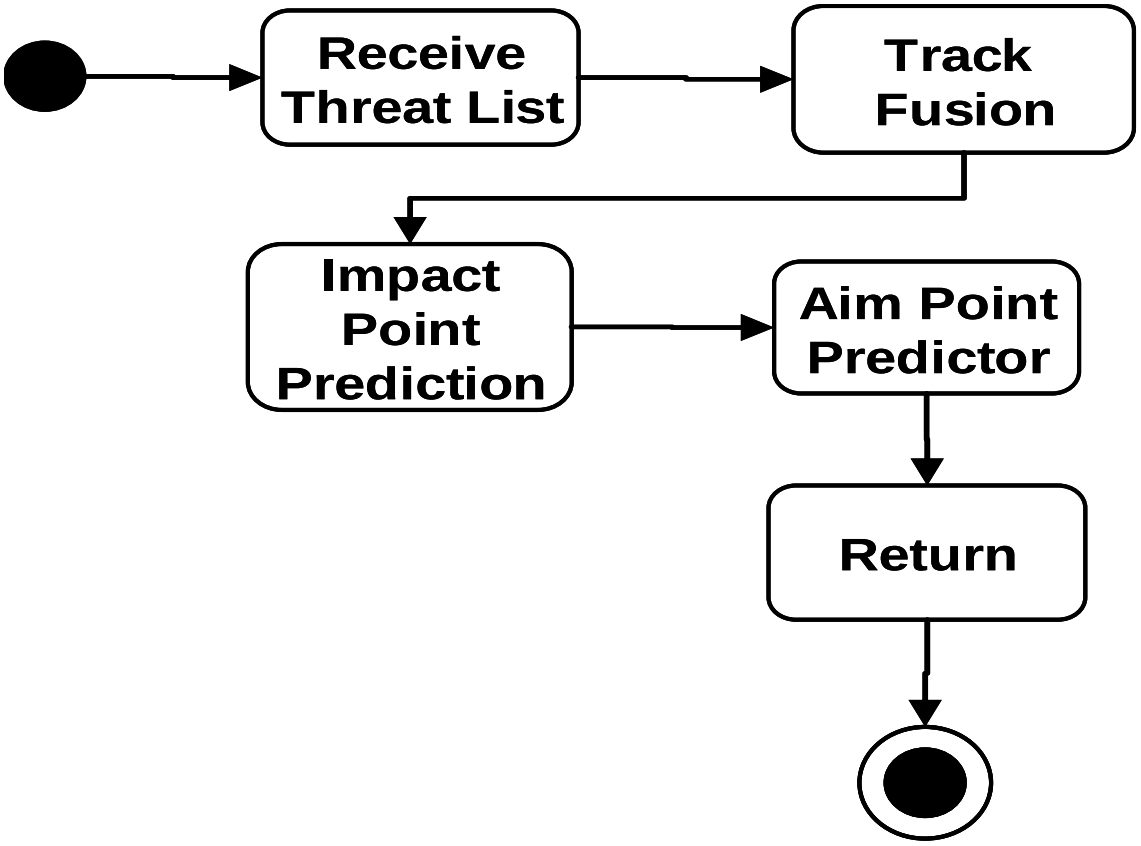

Figure 19. Track

Figure 20 shows the process logic a BM executes upon receiving a threat track list for weapons assignment.

Use Case 3: *Assign Weapon*

Goal in Context: Identify the best weapon system to destroy the threat

Scope & Level: A primary task of a BM

Preconditions: Detect and track tasks in the kill chain has successfully completed

Success End Condition: Identify weapon to destroy the identified threat missiles.

Failed End Condition: A weapon system is not identified

Primary Actor: BM

Trigger: Receive a weapons assignment message

MAIN SUCCESS SCENARIO

1. Receive Launch message

2. Verifies source of message

3. Validate request parameters

4. Monitor QoS requirements

5. Identify available resources

6. Target weapon pairing

7. Return result

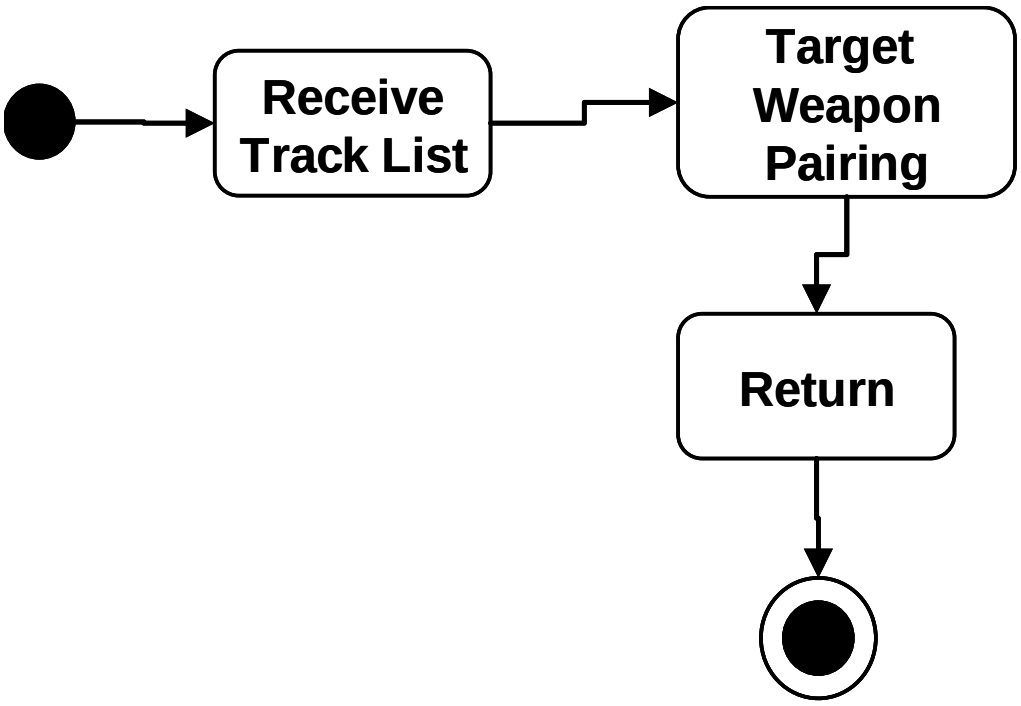

Figure 20. Assign Weapon

Figure 21 shows the process logic a BM executes upon receiving a threat track list with weapons assignment solution to conduct engagement planning.

Use Case 4: *Engage*

Goal in Context: Assigned weapon launches interceptor

Scope & Level: This is a primary task of the TCA

Preconditions: TCA has been initialized

Success End Condition: TCA assigns Launch to appropriate (based on message) weapon or responds to caller that no weapon is available

Failed End Condition: TCA fails to assign weapon or report that there is a problem in launching to caller **Primary Actor:** TCA

Trigger: Receive Launch from superior

MAIN SUCCESS SCENARIO

1. Receive Launch message

2. Verifies source of message

3. Validate request parameters

4. Monitor QoS Requirements

5. Build engagement plan

6. Send plan

7. Monitor QoS Requirements

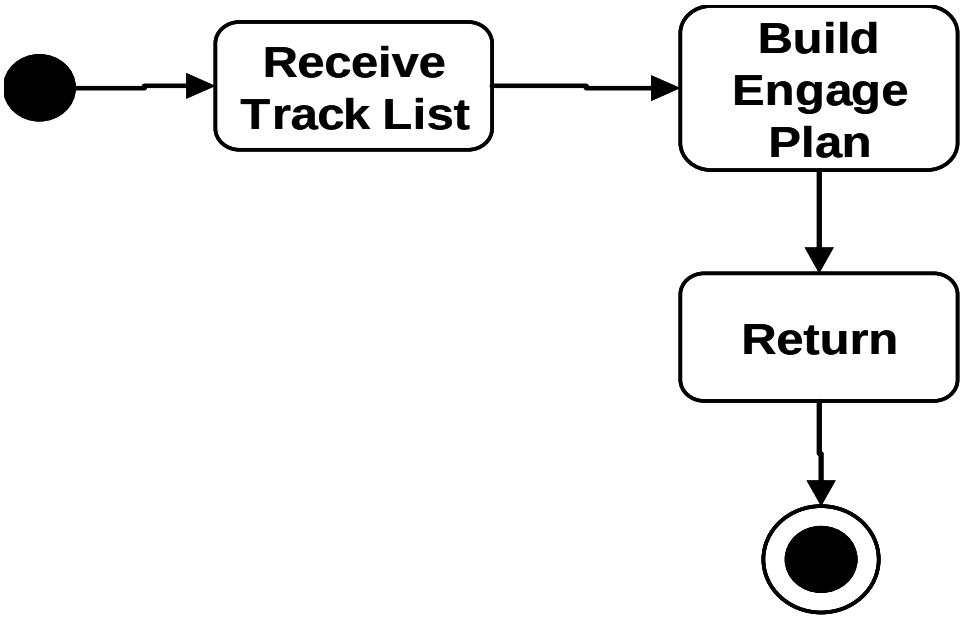

Figure 21. Engage

Figure 22 shows the process logic a BM executes upon receiving an assess kill message.

Conventional battle managers follow a duty cycle commonly referred to as a *kill chain* [9] consisting of the following tasks: detect, track, assign weapon, engage, and assess kill. The kill chain begins when a sensor reports an object to a BM agent. The agent continues to track the object while determining if the object poses a threat, and if the object does pose a threat, assigns an available interceptor to destroy it. After the firing of the interceptor, the BM agent continues to monitor and assess the engagement; if the initial interceptor fails to destroy the threat missile and the shot doctrine used dictates a second shot (e.g. shoot-look-shoot policy) the weapon system re-engages the threat with updated target information.

Use Case 5: *Assess Kill*

Goal in Context: Determine correctly the result of an engagement.

Scope & Level: A primary task of the TCA

Preconditions: TCA has been initialized

Success End Condition: TCA returns a correct assessment of an engagement

Failed End Condition: TCA fails to return a correct assessment of an engagement

Primary Actor: TCA

Trigger: Receive Launch from superior

MAIN SUCCESS SCENARIO

1. Receive Launch message

2. Verifies source of message

3. Validate request parameters

4. Monitor QoS Requirements

5. Report engagement result

6. Return result to caller

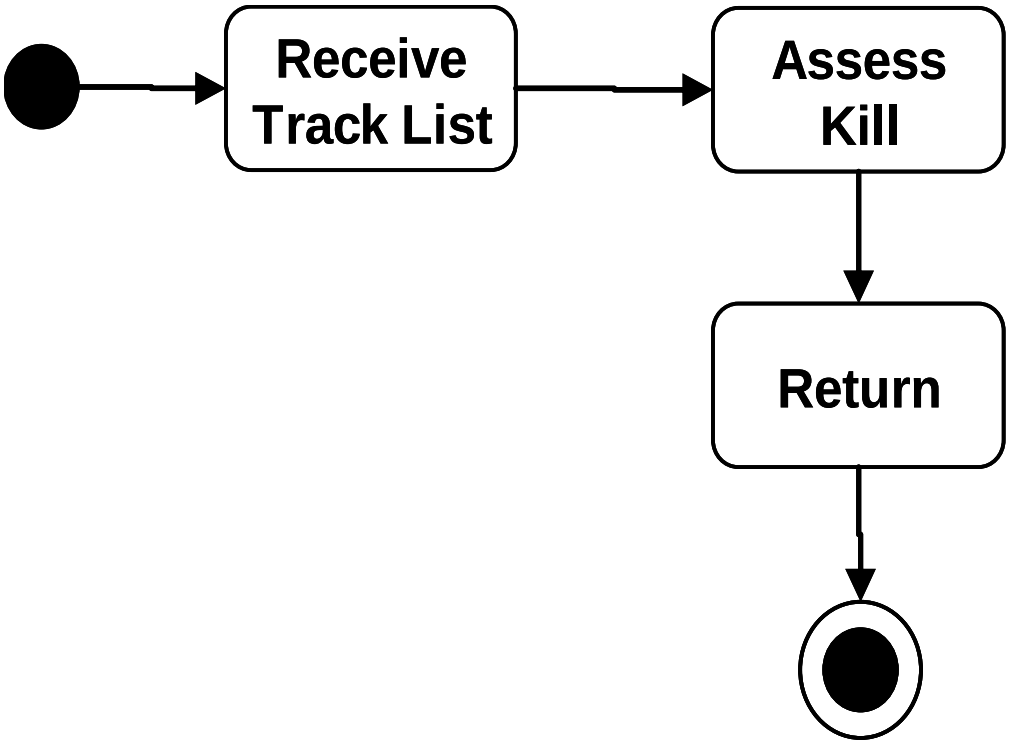

Figure 22. Asses Kill

Figure 23 shows the process logic a BM executes upon first receiving an initialization message followed some time later by an assign weapon message.

In addition to the possibility of a weapons system missing a target there exists the possibility that a BM has no weapons systems available for assignment. In this situation the BM alerts its superior so that an alternative BM can be chosen for the mission; it is customary in military operations to have this built in to the plan and therefore the engage task would have planned to have a number of weapons systems and BM's on stand-by for these type of circumstances. After completion of the kill assessment the duty cycle repeats.

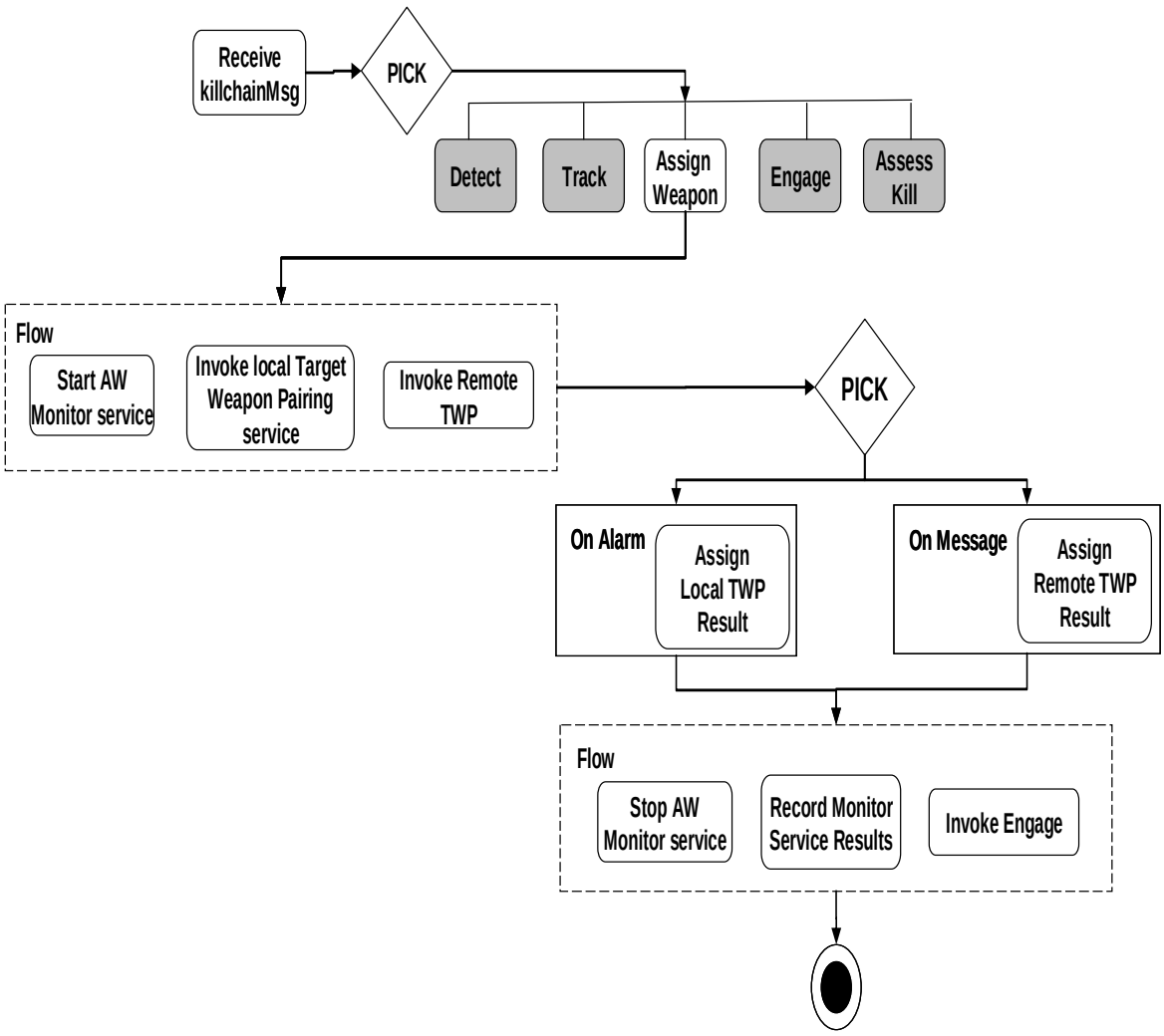

Figure 23. Assign Weapon Process

C. WEB SERVICES FOR BATTLE MANAGEMENT

We now specify a conventional BM as a service in a SoA by specifying the kill chain as a periodic process that is the main orchestrator of a BPEL process decorated with QoS, MOE, and MOP extensions specified in [30]. Selecting the participating partner services of the main kill chain is based on the client QoS and MOE parameters.

1. **Target Association Service (TA):** Begins the kill chain when it receives a list of potential tracks of the threat missile from the sensor net as reported by radars, using a track association algorithm to identify the track objects reported by these sensors.

2. **Track Correlation Service (TC):** Uses a correlation algorithm to compare the reduced track list against a known threat database to classify the missile. If the track object does not match, but observed measurements (e.g. its velocity is in the range of a ballistic missile) makes it suspicious, it is marked suspicious and assigned to additional sensors for observation. All others are logged for offline analysis.

3. **Track Fusion Service (TF):** Track objects gathered thus far are used to create an enhanced description.

4. **Impact Prediction Point Service (IPP):** Predicts the impact point of the threat missile.

5. **Aim Point Predictor Service (APP):** Computes an aiming point for each track object.

6. **Target Weapon Pairing Service (TWP):** Computes the most appropriate weapons systems to engage the threats.

7. **Engagement Planner Service (EP):** Output from TWP and information from the Operations Plan (OPLAN) is used by EP to design, issue, and notify all parties of the plan to destroy the threat missiles.

8. **Assess Kill Service (AK):** Assess battle damage using the sensor net to complete the entire kill chain cycle.

The detect task is composed of the TA and TC services. The track task is composed of TF, IPP, and APP services. The assign weapons task is composed of TWP service. Engage task is composed of an EP service and the assess kill task consists of the AK service. We list the QoS, MoP and MoE parameters of each of the eight services and the five tasks in Table 5. . Figure 24 shows the detect task as a composition of the selected TA service and the TC service.

```
<portType name="DetectPT">
  <operation name="DetectProcess">
    <mopList>
      <mop name="ExecutionTime" value="12sec"/>
      <mop name="Accuracy" value=".998"/>
    </mopList>
    <moeList>
      <moe name="DetectTargetInBOOST" value="null"/>
    </moeList>
    <input message="tns:DetectMsgRequest"/>
    <output message="tns:DetectMsgResponse"/>
  <operation>
</portType>
```

Figure 24. Detect Composition WSDL

Task/Service	QoS	MoP	MoE
Kill Chain	Availability Reliability	Execution Time Accuracy	Kill threat prior keep out range
Detect	Availability Reliability	Execution Time Accuracy	Detect target in boost
Track	Availability Reliability	Execution Time Accuracy	Monitor track without loss of contact
Assign Weapon	Availability Reliability	Execution Time Accuracy	Assign best weapon available
Engage	Availability Reliability	Execution Time Accuracy	Create best mission to destroy threat and monitor BDA
Assess Kill	Availability Reliability	Execution Time Accuracy	Assign best sensor to conduct BDA
Track Association (**TA**)	Availability Reliability	Execution Time Accuracy	Identify and assoc correct number of tracks with its source
Track Correlation (**TC**)	Availability Reliability	Execution Time Accuracy	ID track Objects as threat
Track Fusion (**TF**)	Availability Reliability	Execution Time Accuracy	Enhance threat object by fusing data from multiple

			sensors
Impact Point Prediction (**IPP**)	Availability Reliability	Execution Time Accuracy	Determine IPP within 10 m^2
Aim Point Prediction (**APP**)	Availability Reliability	Execution Time Accuracy	Determine Aim Point within 100 cm^2
Target Weapon Pairing (**TWP)**	Availability Reliability	Execution Time Accuracy	Best weapon to kill target
Engage Planner (**EPS**)	Availability Reliability	Execution Time Accuracy	Best plan to destroy target prior to keep out range
Assess Kill (**AK**)	Availability Reliability	Execution Time Accuracy	Best sensor to conduct BDA

Table 5. QoS, MOP, MOE

```
<operation name="AssocTrackList">
  <mopList>
    <mop name="ExecutionTime" value="5sec"/>
    <mop name="Accuracy" value=".999"/>
  </mopList>
  <moeList>
    <moe name="AssocTrackToSource" value="null"/>
  </moeList>
  <input message="tns:TrackAssocMsgRequest"/>
  <output message="tns:TrackAssocMsgResponse"/>
```

```
  <operation>
</portType>
```

Figure 25. Track Association WSDL

```
<portType name="TrackCorrPT">
  <operation name="CorrTrackList">
    <mopList>
      <mop name="ExecutionTime" value="5sec"/>
      <mop name="Accuracy" value=".999"/>
    </mopList>
    <moeList>
      <moe name="IDThreatGivenThreat" value="NULL"/>
    </moeList>
    <input message="tns:TrackCorrMsgRequest"/>
    <output message="tns:TrackCorrMsgResponse"/>
  <operation>
</portType>
```

Figure 26. Track Correlation WSDL

Finally, we show the WSDL of a complete kill chain that is composed of the higher level tasks detect, track, assign weapon, engage, and assess kill which are themselves composed of the atomic level services described earlier.

```
<portType name="KillChainPT">
  <operation name="killThreat">
    <mopList>
      <mop name="ExecutionTime" value="27sec"/>
      <mop name="Accuracy" value=".95870"/>
    </mopList>
    <moeList>
      <moe name="Defend_Asset" value="NULL"/>
    </moeList>
    <input message="tns:KillChainMsgRequest"/>
    <output message="tns:KillChainMsgResponse"/>
    <fault name="Fail" message="FailNotice:"/>
  <operation>
</portType>
```

Figure 27. Kill Chain WSDL

In each composition, instance execution time was the MOP used to select a service.

One of the two extensions necessary to orchestrate C4ISR services is the need for QoS sensitivity, for which we use the light-weight Q-WSDL extension in [31]. In particular, we use the Operational Latency class where execution times of every operation are specified. The second is the use of the shadow pattern of [32] that specifies exception handling. We use message types, messages and services with the standard notations of '*' for zero or more repetitions, '?' for zero or one repetitions, and '+' for one or more repetitions.

1. Message Types

Table 6 and Table 7 list sample basic and complex WSDL data types [14] used in exchanged messages.

Type Name	Primitive	Example
myId	Long Int	123456789245
sensorID	Long Int	454656736363
Availability	Boolean	Yes/No
weaponName	String	THHAD, AEGIS, …
sensorName	String	FBX, SBX, …
ammoStatus	String	Green, Red, Yellow
timeToEngage	Duration	P0y0m0dt0h0m3s
dateTimeGroup	DateTime	2007-05-31T13:20:00-05:00
Hostile	Boolean	Yes/No
Latitude	Long Int	765468642222
Longitude	Long Int	367463823982
Velocity	Long Int	645646455467
Acceleration	Long Int	832678326864

Table 6. Basic Types of Message elements

Type Name	Type Structure	Example
OPORD	xmlns:OPORD=”http://swe.nps.edu/BMDS/OPORD	opord20080129
Track	XmlNS=URI#trackType	Track Structure
Weapon	XmlNS=URI#weaponType	Weapon structure
Sensor	XmlNS=URI#sensorType	Sensors structure
Sca	XmlNS=URI#scaType	sca structure
Tca	XmlNS=URI#tcaType	tca structure
QoS	XMlNS=URI#qwsdl:operationType [31]	QoS Structure
Bond	XmlNS=URI#Time	$3.5 Cred 1
Turing test	Image	10101..01

Table 7. Complex types of message elements

2. Messages

A sample, assignWeaponMsg, is shown in Listing 27. Other domain-specific messages are listed in Table 8. below with their definitions. Similarly Listing 28 and

Listing 29 show other control messages. Finally, Listing 30 and Listing 31 describe some services provided by the TCA and other third parties.

1. **<message** name="**AssignWeaponMsg**">
2. <part name="ID" element="message ID"/>
3. <part name="Track" element="string"/>
4. <part name=”OPORD” element=”OPORD”/>
5. <part name="DATE" element="Time"/>
6. <part name="QOS" element="QoSType"/>
7. <part name=”surety” element=”Bond”/>
8. <part name=”Ack” element=”wantAck”/>
9. <part name=”Sign” element=”PKISignature/>*
10. <part name=”RTT reply” element=”Turing test
11. **</message>**

Listing 27. WSDL AssignWeaponMsg

Message Type	**Utility**
detectMsg	Kill chain task to associate tracks with a source and determine if the track is a threat
trackMsg	Kill chain task to fuse track information, determine the threat impact point, and calculate an aim point
assignWeaponMsg	Kill chain task to assign the most appropriate weapon to negate a know threat
engageMsg	Kill chain task to build an enagagement plan to

	deafeat a threat
assessKillMsg	Kill chain task to monitor engagement and report Battle Damage Assessment
launchInterceptorMsg	Command to Launch an interceptor
cancelLaunchMsg	Command to cancel a previous launch command
weaponHSMsg	Command to return the health and status of all weapons

Table 8. Types of Messages

```
1. <message name="initializeBMMsg">
2. <part name="id" element="long"/>
3. <part name=”OPORD0129312008” element=”OPORD”
4. </message>
5. <message name="deRequisitionMsg">
6. <part name="ID" element="long"/>
7. </message>
```

Listing 28. WSDL Application Data

```
1. <message name="FailNotice">
2. <part name="Date" element="dateTime"/>
3. <part name="ID" element="long"/>
4. <part name="ERROR" element="string"/>
5. <part name="Sign_PKI" element="string"/>
6. </message>
```

7. **<message** name="**LaunchInterceptorReciept**">
8. <part name="DATE" element="dateTime"/>
9. <part name="ID" element="long"/>
10. <part name="comment" element="string"/>
11. <part name="Sign_PKI" element="string"/>
12. **</message>**

Listing 29. WSDL Control Data

1. **<portType** name="**assignWeaponPT**">
2. <operation name="assignWeapon">
3. <input message="tns:assignWeaponMsg"/>
4. <output message="tns:assignWeaponReciept"/>
5. <faultname="faultassignWeapon "message="tns:FailNotice" / >
6. </operation>
7. **</portType>**

Listing 30. WSDL Port Type Specs for BM services

1. **<portType** name="**monitorServicePT**">
2. <operation name="monoitor">
3. <input message="tns:startMsg"/>
4. <output message="tns:StartNotificationMsg"/>
5. <fault name="monitorfault"
6. message="tns:FailNotice"/>
7. </operation>

8. **</portType>**

9. **<portType** name="**timerPT**">

10. <operation name="startTimer">

11. <:input message="tns:startMsg"/>

12. </operation>

13. **</portType>**

Listing 31. WSDL Port Type for C2 Third Party Services

D. OPERATIONS ORDER (OPORD)

An OPORD is "a directive issued by a commander to subordinate commanders for the purpose of effecting the coordinated execution of an operation."[24] The OPORD is a vital document in ballistic missile defense as it explains in detail the responsibilities of all systems. The OPORD, at a minimum, contains unit task organization and the five paragraphs of (1) Situation (2) Mission (3) Execution (4) and Service Support (5) Command and Signal.

In traditional land warfare combat commanders issue their orders to subordinate commanders who in turn prepare and issue orders to their subordinates until each combatant in every unit knows his or her mission and the mission of those two levels up the chain of command. The initial OPORD of nearly all campaigns are routinely more detailed and well thought out than subsequent OPORDs. This tendency is a direct reflection of the amount of time available to plan prior to hostilities beginning. For the initial order, units may have days, weeks, and even months to plan and issue the orders. Once hostilities begin, the time to plan generally decreases and makes the development, issuance and coordination of plans more difficult, in addition to reducing timelines to days or hours.

In missile defense the timelines are significantly shorter than traditional land warfare combat scenarios discussed above. In the missile defense domain timelines can be in the range of several minutes to as little as 30 seconds.

With such short timelines we look to perform autonomous execution of missile defense engagements where we remove the human from the loop. For this reason the OPORD must be designed to be read and "understood" by computers; we accomplish this in our case study by constructing our OPORDs using the Resource Description Framework (RDF) [33]. RDF is a W3C Recommendation for describing Web resources and is designed to be processed by computers. We show in Listing 32 below our OPORD written in RDF/XML for the scenario described above and pictured in Figure 17. The RDF provides the means to describe the complex structure of the OPORD so that it can be understood by the participating BMs. In Listing 32, lines 40, 44, 48, 52, and 56 show the five minimum essential paragraphs of an OPORD as defined in [24]. Each of the five paragraphs is a property that has a reference to a resource containing information about the particular property. As an example we show at line 47 the property OPORD:MISSION has a reference to a resource containing information about the TCA's MISSION; the text of an actual mission for TCA22 is in bold. It is certain that some of the other paragraphs have subgraphs and each of those can be defined by a value or as in the case of the OPORD MISSION a reference to another resource.

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:rdf=
3.  "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5.  xmlns:OPORD="http://swe.nps.edu/BMDS/OPORD#">
//The rdf description element that describes our resource
// OPORD
6.  <rdf:Description rdf:about=
7.  "http://swe.nps.edu/BMDS/OPORD/opord20080129">
8.  <OPORD:classification>
9.  <rdf:Alt>
10.     <rdf:li>UNCLASS</rdf:li>
11.     </rdf:Alt>
12.     </OPORD:classification>
13.     <OPORD:CopyNumOfNumCopies>1 of 100
14.     </OPORD:CopyNumOfNumCopies>
15.     <OPORD:issuingHQ>TCA21</OPORD:issuingHQ>
16.     <OPORD:placeofIssue>452143</OPORD:placeofIssue>
17.     <OPORD:DTGSignature>012920080100
18.     </OPORD:DTGSignature>
```

19. <OPORD:MsgRefNum>012920080245
20. </OPORD:MsgRefNum>
21. <OPORD:OrderNumber>01292008-45
22. </OPORD:OrderNumber>
23. <OPORD:codeName>Butkus</OPORD:codeName>
24. <OPORD:references>
25. <rdf:Seq>
//Reference to the SCA's Initiating OPORD Code Name Lambert
26. <rdf:li rdf:resource=
27. "http://swe.nps.edu/BMDS/documents/OPORD/Lambert">
28. </rdf:li>
//Reference to the RCA2's Initiating OPORD Code Name HAM
29. <rdf:li rdf:resource=
30. "http://swe.nps.edu/BMDS/documents/OPORD/Ham">
31. </rdf:li>
32. </rdf:Seq>
33. </OPORD:references>
34. <OPORD:timeZoneUsed>ZULU
35. </OPORD:timeZoneUsed>
//Task Organization is defined by the resource URI below
36. <OPORD:taskOrginization
37. rdf:resource=
38. "http://swe.nps.edu/BMDS/OPORD/opord20080129
39. /AnnexATaskO"></OPORD:taskOrginization>
//The first of the minimum essetial elements of the five
//Paragraph operations order; SITUATION
40. <OPORD:SITUATION
41. rdf:resource=
42. "http://swe.nps.edu/BMDS/OPORD/opord20080129
43. /Situation"></OPORD:SITUATION>
//The second of the minimum essetial elements of the five
//Paragraph operations order; MISSION
44. <OPORD:MISSION
// The URI at line 46 is a reference to the document that
//contains the following mission statement for this OPORD
//MISSION: 060004282008 (Z) TCA22 forces Defend
//assets according to the Priority Defended Assets List
//(PDAL) against anticipated ballistic missile attacks
//within region.
45. rdf:resource=
46. "**http://swe.nps.edu/BMDS/OPORD/opord20080129**
// The URI at line 48 is a reference to the document that
//contains the next Higher level of commands (RCA2 in
this scenario) mission statement for this OPORD
//MISSION: 060004282008 (Z) RCA2 forces Defend

//assets according to the Priority Defended Assets List
//(PDAL) against anticipated ballistic missile attacks
//within region.
47. <OPORD:HIGHERMISSION
48. rdf:resource= " http://swe.nps.edu/BMDS/documents
/OPORD/Ham.MISSION
//The third of the minimum essetial elements of the five
//Paragraph operations order; EXECUTION
49. <OPORD:EXECUTION
50. rdf:resource=
51. "http://swe.nps.edu/BMDS/OPORD/opord20080129
52. /Execution"></OPORD:EXECUTION>
//The fourth of the minimum essetial elements of the five
//Paragraph operations order; SERVICE SSUPPORT
53. <OPORD:SERVICESUPPORT
54. rdf:resource="http://swe.nps.edu/BMDS/OPORD
55. /opord20080129/ServiceSupport">
56. </OPORD:SERVICESUPPORT>
//The fifth of the minimum essetial elements of the five
//Paragraph operations order; COMMANDANDSIGNAL
57. <OPORD:CMDSIGNAL
58. rdf:resource="http://swe.nps.edu/BMDS/OPORD
59. /opord20080129/CommandSignal">
60. </OPORD:CMDSIGNAL>
61. <OPORD:CDRSNAMERANK>COOKGEN
62. </OPORD:CDRSNAMERANK>
63. <OPORD:AUTHENNAMEPOS>PULFORD2IC
64. </OPORD:AUTHENNAMEPOS>
//ANNEXES A-Z OF THE OPORD EACH IS A RDF
//RESOURCE WHOS DESCRIPTION IS FOUND AT
//THE APPROPRIATE URI
65. <OPORD:ANNEXES>
66. <rdf:Seq>
67. <rdf:li rdf:resource="http://swe.nps.edu/BMDS/OPORD
68. /opord20080129/AnnexATaskO"></rdf:li>
69. …<rdf:li rdf:resource="http://swe.nps.edu/BMDS/OPORD
70. /opord20080129/AnnexZDistro"></rdf:li>
71. </rdf:Seq>
72. </OPORD:ANNEXES>
73. <OPORD:DISTROrdf:resource="http://swe.nps.edu
74. /BMDS/OPORD/opord20080129/AnnexZDistro">
75. </OPORD:DISTRO>
76. </rdf:Description>
77. </rdf:RDF>

Listing 32. Operations Order

E. BPEL ORCHESTRATION OF BATTLE MANAGER

In this section we specify the TCA using BPEL [34], where the TCA Assign Weapon Process invokes necessary local and remote services. In Listing 33 the TCA is activated upon receiving an initialization message, which includes operations order shown in Listing 19, from the RCA, to establish the organizational structure thereby creating the chain of command for the SCA, RCA, TCA and weapons and sensors nets. Once the TCA process completes initialization it blocks waiting for one of the predefined messages detect, track, assign weapon, engage, assess kill, cancel launch, switch mode, or other commands from its RCA. In our scenario, TCA21 receives the assign weapon message from RCA2 lines 47-49 of Listing 33.

As specified in lines 50-67, in response to an assign weapon message, the TCA invokes a local monitoring service, to record information on the executing services and the assign weapons task in its entirety and the remote target weapon pairing (RTWP) service algorithm. At line 110 the process invokes a local asynchronous Local Target Weapon Pairing (LTWP). This service acts as a shadow [32] to the RTWP service. If the assign weapon process does not receive a result from the RTWP within 10 seconds, an alarm is triggered in line 40 alerting the process to use the result from the LTWP service for the rest of the task.

Upon receiving the TWP result the process invokes the Engage task line 148 with the result and waits ten seconds for a callback line 156 signaling that the engage task has been initiated after which the process invokes the stop monitor and records the QoS, MOP, and MOE results. Finally, if the process does not receive the callback message from the engage task, the process invokes the warning callback line 136 to the calling client signaling that the task completed, but there is no evidence that the engage task received the results or has begun execution.

1. **<?xml version=”1.0” encoding=”UTF-8”?>**
2. <:process
3. xmlns:AW=”http://swe.nps.edu/bmds/services/AW”
4. :ENG=”http://swe.nps.edu/bmds/services/ENGAGE”
5. …
6. <import importType=”http://schemas.xmlsoap.org/wsdl/”

location=”WSDL/AwMonitor.wsdl”
ns=”http://swe.nps.edu/BMDS/service/awonitor/”>
7. …<partnerLinks>
8. <:partnerLink myRole=
9. “awService” name=”assignWeapon” partnerLinkType=”AW:awLT”
partnerRole=”AwCustomer”/>
10. …</:partnerLinks>
11. <:variables>
12. <:variable messageType=”AW:AWMsg” name=”AWMsg”/>
13. …</:variables>
14. <:flow>
15. <:links>
16. …</:links>

// lines 17 – 19 Receive Assign Weapon from RCA2
// execute the targetWeaponPairing operation

17. <:receive createInstance=”yes”
18. name=”ReceiveAWMsg”
19. operation=”targetWeaponPairing” partnerLink=”assignWeapon”
portType=”AW:assignWeaponPT” variable=”AWMsg”>
20. <:sources>
21. <:source linkName=”L3”/>
22. </:sources>
23. </:receive>
24. <:pick name=”PickLocalOrRemoteResult”>
25. <:targets>
26. <:target linkName=”L9”/>
27. </:targets>
28. <:sources>
29. <:source linkName=”L4”/>
30. </:sources>

// line 31-35 if receive the callback message assign the
// results to engageMsg1

31. <:onMessage operation=”TwpCallback” partnerLink=”remoteTwpLT”
portType=”rtwp:TwpCallbackPT” variable=”remoteTwpResponseMessage”>
32. <:assign name=”AssignRemoteTWPResult”>
33. <:copy>
34. <:from part=”result” variable=”remoteTwpResponseMessage”/>
35. <:to part=”awResult” variable=”engageMsg1”/>
36. </:copy>
37. </:assign>
38. </:onMessage>
39. <:onAlarm>
40. <:for>PT10S</:for>
41. <:assign name=”AssignLocalTWPResult”>
42. <:copy>

43. <:from part=”result”
44. variable=”localTWPResponseMessage”/>
45. <:to part=”awResult” variable=”engageMsg1”/>
46. </:copy>
47. </:assign>
48. </:onAlarm>
49. </:pick>

// lines 50-67 CONCURENTLY Call the remote target
// weapon pairing algorithms and the moitoring service

50. <:flow name=”FlowStartMon_RemoteTWP”>
51. <:targets>
52. <:target linkName=”L3”/>
53. </:targets>
54. <:sources>
55. <:source linkName=”L5”/>
56. </:sources>
57. <:links>
58. <:link name=”L1”/>
59. <:link name=”L2”/>
60. </:links>
61. <:invoke inputVariable=”startMonitorRequestMessage” name=”InvokeAWMonitorService” operation=”startMonitor” partnerLink=”monitorLT” portType=”awmon:startMonitorPT”>
62. <:targets>
63. <:target linkName=”L1”/>
64. </:targets>
65. </:invoke>
66. <:invoke inputVariable=”remoteTwpRequestMessage” name=”InvokeRemoteTWP” operation=”Twp” partnerLink=”remoteTwpLT” portType=”rtwp:remoteTwprequestPT”>
67. <:targets>
68. <:target linkName=”L2”/>
69. </:targets>
70. </:invoke>
71. <:assign name=”AssignMonitorParams”>
72. <:sources>
73. <:source linkName=”L1”/>
74. </:sources>
75. <:copy>
76. <:from>
77. <:literal>start</:literal>
78. </:from>
79. <:to part=”start” variable=”remoteTwpRequestMessage”/>
80. </:copy>
81. </:assign>

82. <:assign name=”PassTrackList”>
83. <:sources>
84. <:source linkName=”L2”/>
85. </:sources>
86. <:copy>
87. <:from>
88. <:literal>tracklist</:literal>
89. </:from>
90. <:to part=”start”
91. variable=”remoteTwpRequestMessage”/>
92. </:copy>
93. </:assign>
94. </:flow>
95. <:sequence name=”SequenceLocalTWP”>
96. <:targets>
97. <:target linkName=”L5”/>
98. </:targets>
99. <:sources>
100. <:source linkName=”L6”/>
101. </:sources>
102. <:assign name=”PassTrackList”>
103. <:copy>
104. <:from>
105. <:literal>start</:literal>
106. </:from>
107. <:to part=”start” variable=”localTWPRequestMessage”/>
108. </:copy>
109. </:assign>

// line 110 synchronous call to the local target weapon
// paring algorithm

110. <:invoke inputVariable=”localTWPRequestMessage” name=”InvokeLocalTWP” operation=”localTWP” outputVariable=”localTWPResponseMessage” partnerLink=”localTWPLT”
111. portType=”ltwp:localTWPPT”/>
112. </:sequence>
113. <:pick name=”Wait10SecForCallback”>
114. <:targets>
115. <:target linkName=”L8”/>
116. </:targets>

// line 117 waiting for callback from the Engage task that
// is invoked after Assign weapon completes its task.

117. <:onMessage operation=”callback” partnerLink=”Engage” portType=”ENG:engageCallBackPT” variable=”callBackMsg”>
118. <:flow>
119. <:links>

120. <:link name=”L10”/>
121. </:links>
// line 122 – 127 Stop the monitor and record the results
122. <:invoke inputVariable=”stopMonitorRequestMessage” name=”InvokeStopMonitor” operation=”stopMonitor” partnerLink=”StopMonitorLT” portType=”awmon:stopMonitorPT”>
123. <:sources>
124. <:source linkName=”L10”/>
125. </:sources>
126. </:invoke>
127. <:receive name=”RecordMonitorResponseResults” operation=”monitorCallback” partnerLink=”monitorLT” portType=”awmon:monitorCallbackPT” variable=”monitorResponseMessage”>
128. <:targets>
129. <:target linkName=”L10”/>
130. </:targets>
131. </:receive>
132. </:flow>
133. </:onMessage>
134. <:onAlarm>
135. <:until>P10S</:until>
// **line 136 invokes callback alerting the client that while**
// assign weapon completed its task it has not received
// confirmation from the engage task
136. <:invoke inputVariable=”WarningMsg” name=”InvokeWarningCallback”
137. operation=”warningCallback” partnerLink=”assignWeapon” portType=”AW:WarningPT”/>
138. </:onAlarm>
139. </:pick>
// Line 140 Receive the results from the remote target
// weapon pairing algorithm
140. <:receive name=”ReceiveRemoteTWPcallback” operation=”TwpCallback” partnerLink=”remoteTwpLT” portType=”rtwp:TwpCallbackPT” variable=”remoteTwpResponseMessage”>
141. <:targets>
142. <:target linkName=”L6”/>
143. </:targets>
144. <:sources>
145. <:source linkName=”L9”/>
146. </:sources>
147. </:receive>
// line 148 invoke the Engage task of the kill chain
148. <:invoke inputVariable=”engageMsg1” name=”InvokeEngage” operation=”engage” partnerLink=”Engage” portType=”ENG:engagePT”>
149. <:targets>

```
150.    <:target linkName=”L4”/>
151.    </:targets>
152.    <:sources>
153.    <:source linkName=”L7”/>
154.    </:sources>
155.    </:invoke>
//156 receive a callback from the engage task alerting the
// client that task handoff is complete
156.    <:receive name=”ReceiveEngageCallback” operation=”callback”
        partnerLink=”Engage” portType=”ENG:engageCallBackPT”
        variable=”callBackMsg”>
157.    <:targets>
158.    <:target linkName=”L7”/>
159.    </:targets>
160.    <:sources>
161.    <:source linkName=”L8”/>
162.    </:sources>
163.    </:receive>
164.    </:flow>
165.    </:process>
```

Listing 33. The TCA Process

F. EVOLUTION OF OPORDS

As discussed in the previous chapter, the OPORD provides the mission of two higher levels of command and tasks to subordinates. Once the initial order is received the commander must delete the higher level commands mission and add his own. In addition, the commander must replace the tasks to the subordinates with his own. Typical information in the tasks are things such as the defended sector assignment and orientation and asset to be defended in sector (e.g. from the PDAL).

In our scenario the threat ballistic missiles do not attack according to the intelligence estimate sent out in the initial OPORD and RCA issues a FRAGMENTARY ORDER (FRAGO) as part of the AssignWeaponMsg to its subordinate TCAs to be prepared to reorient weapons systems and sensors in line 17. The AssignWeaponMSG would contain those parts of the OPORD that had change; for instance the mission is the

same, but the execution paragraph would task sensors and weapons to reorient in the general direction of the incoming target so that the weapons systems could engage at the earliest opportunity.

G. CONCLUSIONS

In this chapter we have shown how the BPEL with appropriate extensions for MoPs, MoEs and QoS parameters can be used to specify C2BM needs of ballistic missile control.

V. PERFORMANCE FEEDBACK MESSAGES

The performance feedback mechanism is built into the BPEL orchestrations by having a monitoring service that starts a local timer and monitors the assigned MOPs and MOEs when the orchestration runs. Once the orchestration completes a cycle, the monitor returns the orchestration execution time, updates the results for the assigned MOPs and MOEs, and returns a message to the associated recipients. In this chapter, we show the design of a service-oriented performance feedback message written in RDF for a single MOE associated with the execution of a kill chain. The MOE identified for all participating of the kill chains is, *Percent of Defended Asset Locations (DAL) successfully defended*. We show a specification for the performance feedback messages of the kill-chain process that are reported to their associated command agent, that contain the percentage of DAL successfully defended and other necessary information. We show the roll up of the MOE from the TCA up through the RCAs to the highest level command, SCA, in Figure 28.

In Figure 28 each TCA reports the total number of threat missiles destroyed out of the total number of threat missiles engaged within its command to its RCA. TCA21 reports that it has destroyed one of the two threat missiles, while TCA22 reports that it has destroyed all three of the threat missiles that it has engaged. The RCAs then report the total number of threat missiles destroyed out of the total threat missiles engaged within in their command to the SCA. RCA2 reports to the SCA that it has destroyed four of the six threat missiles engaged within its command. The SCA then totals the number of threat missiles destroyed out of the total number of threat missiles engaged within the SCA to produce the MOO—total number of DAL successfully defended, as shown in Figure 28. The MOO is eighteen threat missiles destroyed of the twenty-four threat missiles engaged. For clarity, we make the following assumptions:

- Any threat missile engaged is heading for one single defended asset.
- If the threat missile is destroyed then the defended asset is successfully defended.

- If a threat missile is not destroyed during an engagement the defended asset is destroyed and not successfully defended.

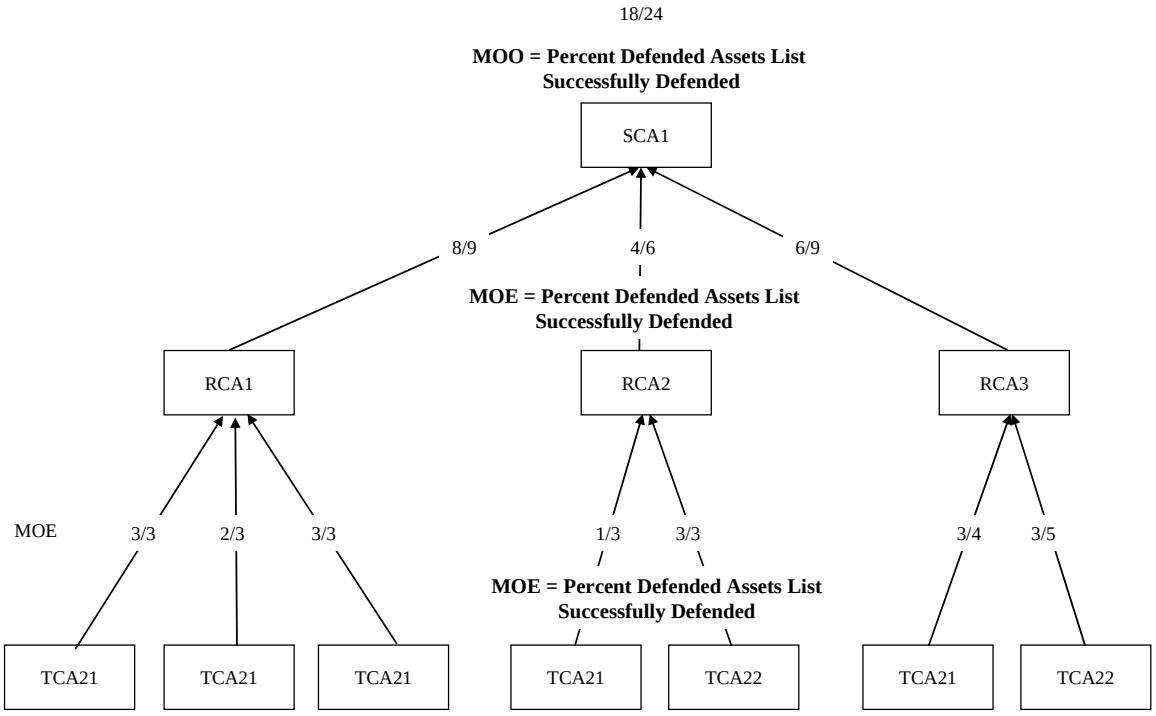

Figure 28. MOE Roll-Up to MOO

We show now our specification of the feedback messages for updating the command agent's MOE and MOO using RDF. We point out that our specification shows only a single MOE. It is likely that several MOEs and MOOs would be included and rolled-up in a similar manner.

A. MESSAGE FORMAT TYPE

We developed an automatic report format to record and update MOO and MOE to the higher level command based on the Spot Report Format from [43] shown below in Figure 29.

Spot Report [SPOTREP]

REPORT NUMBER: S060

GENERAL INSTRUCTIONS: Use to send information to provide timely intelligence or status regarding events

that could have an immediate and significant effect on current planning and operations. Reference: FM 3-20.15,

FM 3-20.98, and FM 3-21.71.

LINE 1 – DATE AND TIME _____ (DTG)

LINE 2 – UNIT _____ (Unit Making Report)

LINE 3 – SIZE _____ (Size of Enemy Unit)

LINE 4 - ACTIVITY _____ (Enemy Activity at DTG of Report)

LINE 5 - LOCATION _____ (UTM or Six-Digit Grid Coordinate

With MGRS Grid Zone Designator of

Enemy Activity or Event Observed)

LINE 6 - UNIT_____ (Enemy Unit)

LINE 7 – TIME_____ (DTG of Observation)

LINE 8 – EQUIPMENT _____ (Equipment of Unit Observed)

LINE 9 - SENDER'S ASSESSMENT_____ (Specific Sender Information)

LINE 10 – NARRATIVE _____ (Free Text for Additional Information

Required for Clarification of Report)

LINE 11 – AUTHENTICATION _____ (Report Authentication)

Figure 29. SPOTREP (From: [43])

The SPOTREP is used by the US Army forces to report "intelligence or status of events that could have an immediate and significant effect on current planning and operations."[3]

Briefly, the information contained in a standard SPOTREP is identified below by line number.

- Line 1 contains the date-time group that the message is created
- Line 2 contains the name of the unit making the spot report
- Line 3 contains the size of the enemy unit being observed
- Line 4 contains a description of the activity of the enemy unit at the time given at line 1
- Line 5 contains the location of the enemy or event using the Military Grid Reference System and preceded by grid zone designator
- Line 6 contains the enemy units name if known
- Line 7 contains the date-time group of the observation
- Line 8 contains information on the enemy equipment
- Line 9 contains the senders Assessment
- Line 10 contains free text
- Line 11 contains message authentication

The SPOTREP format is near suitable to serve as the feedback message for our system, but it does require two modifications similar to those made to the OPORD message, discussed previously in Chapter III.

First we write the SPOTREP message in RDF format to make it machine processable and second we tailor the SPOTREP format to include information pertaining to MOOs and MOEs.

[3] This quote is taken from [43] which reference the following U.S. Army field Manuals: FM 3-20.15, FM 3-20.98, and FM 3-21.71.

We show next in section B the details of our new machine processable SPOTREP message and explain how it directly supports the automatic performance feedback for the kill-chain service.

B. RDF SPOTREP

The header of the RDF SPOTREP contains the namespaces used to build this message. Many of the namespaces in the SPOTREP are reused from other sources to include those created for the OPORD.

```
1: <?xml version="1.0"?>
2: <rdf:RDF xml:lang="en"
3:xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
4: xmlns:bmds=http://swe.nps.edu/bmds/elements/1.0/
5:xmlns:opord=http://swe.nps.edu/bmds/opord/
6:xmlns:geo=http://www.w3.org/2003/01/geo/wgs84_pos#
7: xmlns:dcterms=http://purl.org/dc/terms
8: xmlns:dc=http://purl.org/dc/elements/1.1/
9: xmlns:rep=http://swe.nps.edu/bmds/messages/rep/
10: xmlns:nps=http://swe.nps.edu/bmds/UJTL
11: xml:base="http://swe.nps.edu/bmds/messages/rep/SPOTREP/">
```

Listing 34. SPOTREP Name Spaces

Line 14 of the message is important as it identifies tca22sr2008-01-29T00:00:00-0019:00.htm as the particular instance of a spot report being described.

```
13: <!-- SPOTREP Message A message sent periodically to Alert higher
level command of enemy activity The resource is named by the originator
and the date time group which is used for correlation-->
14: <bmds:Resource rdf:about="tca22sr2008-01-29T00:00:00-19:00.htm">
```

Listing 35. SPOTREP Resource

Line 16 through line 30 describes biographical information about this specific spot report message. Much of the definition is the same as found in the OPORD from Chapter III. We have added an element to identify the issuing agent of the spot report message, that maps to line 2; unit making the report, in Figure 29. The date and time the SPOTREP was created is defined by the <<dc:Created>> tag at line 20 and maps to line 1, DATE and TIME, in Figure 29. Line 24 through 30 of Listing 36 are re-used directly from our OPORD.

```
16: <!-- SPOTREP Message Biography Message used as in the -->
17: <!--Resource biographical information-->
18: <bmds:bio rdf:parseType="Resource">
19:    <dc:title> tca22sr SPOTREP</dc:title>
20:    <dc:dateCreated>2008-01-29T00:00:00-19:00</dc:dateCreated>
21:    <bmds:timeZoneUsed>ZULU</bmds:timeZoneUsed>
22:    <dc:author>Auto Generated</dc:author>
23:    <rep:issueAgent>TCA22</rep:issueAgent>
24:    <bmds:classification>unclassified</bmds:classification>
25:    <bmds:placeIssued>
26:       <geo:Point>
27:       <geo:lat>20.20</geo:lat>
28:       <geo:long>-90.80</geo:long>
29:    </geo:Point>
30:   </bmds:placeIssued>
```

Listing 36. SPOTREP Bio_1

Line 31 through line 48 of Listing 37 continues the biographical information pertaining to this SPOTREP message. Line 31 through line 36 lists all the resources that this message references. In this example SCA1 OPORD is the only reference. An important addition to the SPOTREP for the purposes of automation is the inclusion of a correlation construct shown from line 38 through line 42. The correlation construct lists all previous related spot report messages. The final construct shown below at line 47 is the <<bmds:threatTrackList>>. This is an important construct as it identifies the threat track list that is associated with the SPOTREP.

```
31: <!-- This report is tied directly to the the SCA1 OPORD
          through the use of the references tag -->
32: <opord:references>
33: <rdf:Seq>
34: <rdf:li rdf:resource="http://swe.nps.edu/bmds/opord/SCA1opord20080129">
    </rdf:li>
35: </rdf:Seq>
36: </opord:references>
37: <!-- Correlation construct that associates this message with a
        list containing all previous Spot reports associated with
        this threatTrackList ID-->
38: <rep:correlation>
39: <rdf:Seq>
40: <rdf:li rdf:resource="tca22sr2008-01-29T00:00:00-18:59.htm"></rdf:li>
41: </rdf:Seq>
42: </rep:correlation>
44: <bmds:msgRefNum>Message Reference Number</bmds:msgRefNum>
46: <!-- threatTrackList ID assigned and updated by the battle manager -->
47: <bmds:threatTrackList rdf:resource="http://swe.nps.edu/bmds/threatTrackList">
    </bmds:threatTrackList>
48: </bmds:bio>
```

Listing 37. SPOTREP Bio_2

Listing 38 below identifies all of the internally related resources for the particular instance of the spot report. Consequently, it points to the hit ratio, MOE, and MOO associated with this spot report. We further define these resources in Listing 39.

```
49: <!-- These are the internal resources that describe the individual
        pieces of the SPOT REPORT correlated to a specific
        threatTrackList ID assigned and updated by the battle
        manager, Specifically the MOO, MOE, and hit ratio.
        Each is described in more detail below-->
51: <rep:related rdf:resource="hitRatio" />
52: <rep:related rdf:resource="MOE" />
53: <rep:related rdf:resource="MOO" />
54: <rep:related rdf:resource="" />
56: </bmds:Resource>
```

Listing 38. SPOTREP Internal Resources

As stated above, we now describe the hit ratio and measures information pertaining to the spot report. This information maps to line 9, sender's assessment, of Figure 29. Line 57 through 67 describes the constructs for the MOE and MOO respectively. In our example the MOE and MOO have the same measure called "percent" and the same measure description called "Of protected DAL locations, successfully defended". The purpose for identically named MOE and MOO is so that all levels of commands can use the same message format to roll-up the MOEs to the highest level command MOO. We demonstrated this previously in Figure 28. The last construct, hit ratio, in Listing 39 is shown from line 69 through line 72. The hit ratio consists of the number of targets engaged; shown at line 70 over the number of targets destroyed; shown at line 71. In this example the message is sent from TCA22 to its regional commander RCA2. The hit ratio indicates that the target has been engaged, but the assessment to this point shows that the target has not been destroyed.

Line 74, the final line of Listing 39, is the closing RDF tag for the entire SPOTREP message.

```
57: <!-- This describes the MOE used in this particular
            engagement if applicable.-->
58: <rdf:Description rdf:about="MOE">
59: <nps:measure>Percent</nps:measure>
60: <nps:measureDescr>Of protected DAL locations, successfully
                          defended.</nps:measureDescr>
61: </rdf:Description>
62:
63: <!-- This describes the MOO used in this particular engagement if
applicable.  -->
64: <rdf:Description rdf:about="MOO">
65: <nps:measure>Percent</nps:measure>
66: <nps:measureDescr>Of protected DAL locations, successfully
defended.</nps:measureDescr>
67: </rdf:Description>
68: <!-- This is the definition of a hit ration.  -->
69: <rdf:Description rdf:about="hitRatio">
70: <nps:numTgtEngaged>1</nps:numTgtEngaged >
71: <nps:NumTgtDestroyed>0</nps:NumTgtDestroyed >
72: </rdf:Description>
73:
74: </rdf:RDF>
```

Listing 39. SPOTREP MOE, MOO, Hit_Ratio

As the SPOTREP is now in a machine-processable format, our monitor services can be used to update messages. Message-delivery services can report the updated information to the appropriate clients and users, who use the information to update service WSDLs and update MOEs and MOOs.

We next show in Figure 30 through Figure 33 below the entire RDF graph for the SPOTREP message described in Listing 34 through Listing 39.

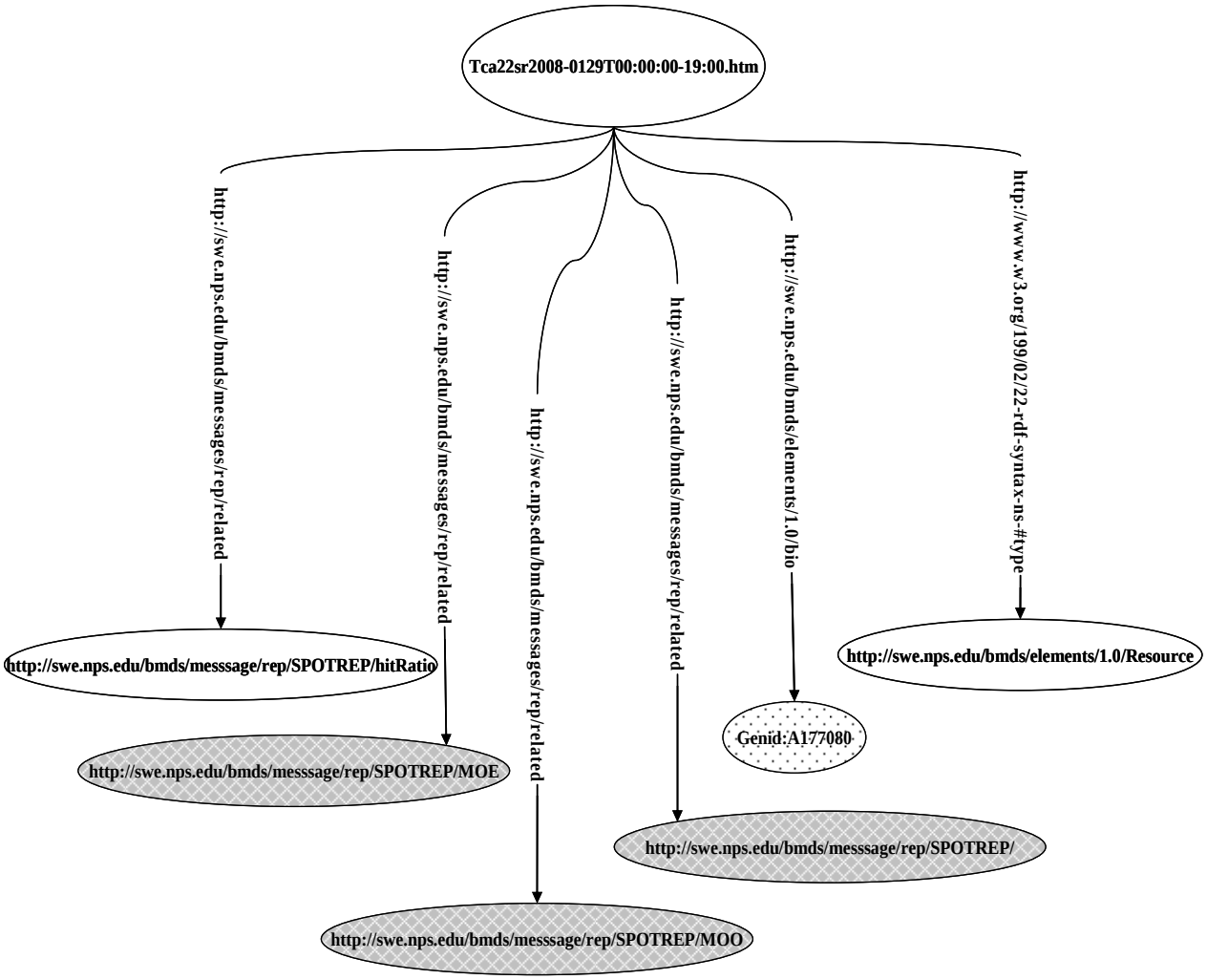

Figure 30. SPOTREP RDF Graph describing Spot Report Tca22sr

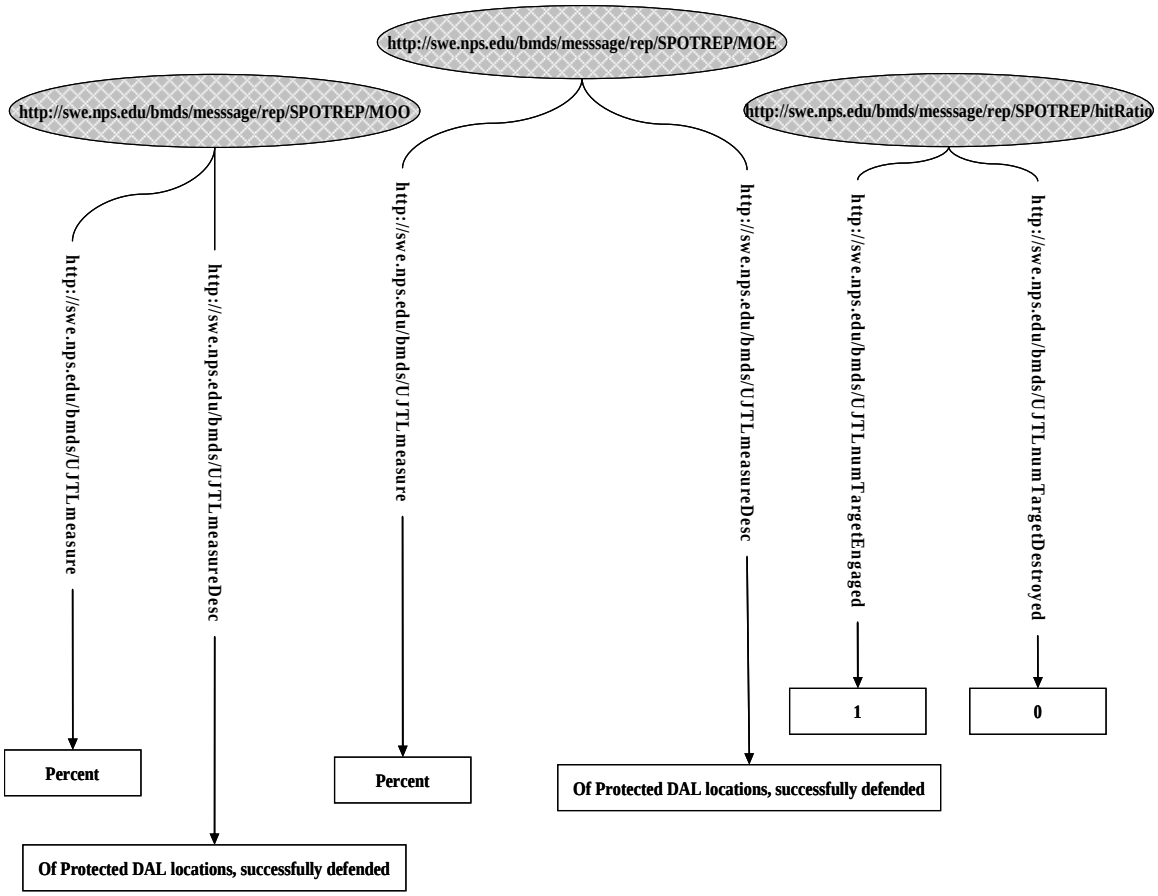

Figure 31. SPOTREP RDF Graph MOO, MOE, Hit_Ratio

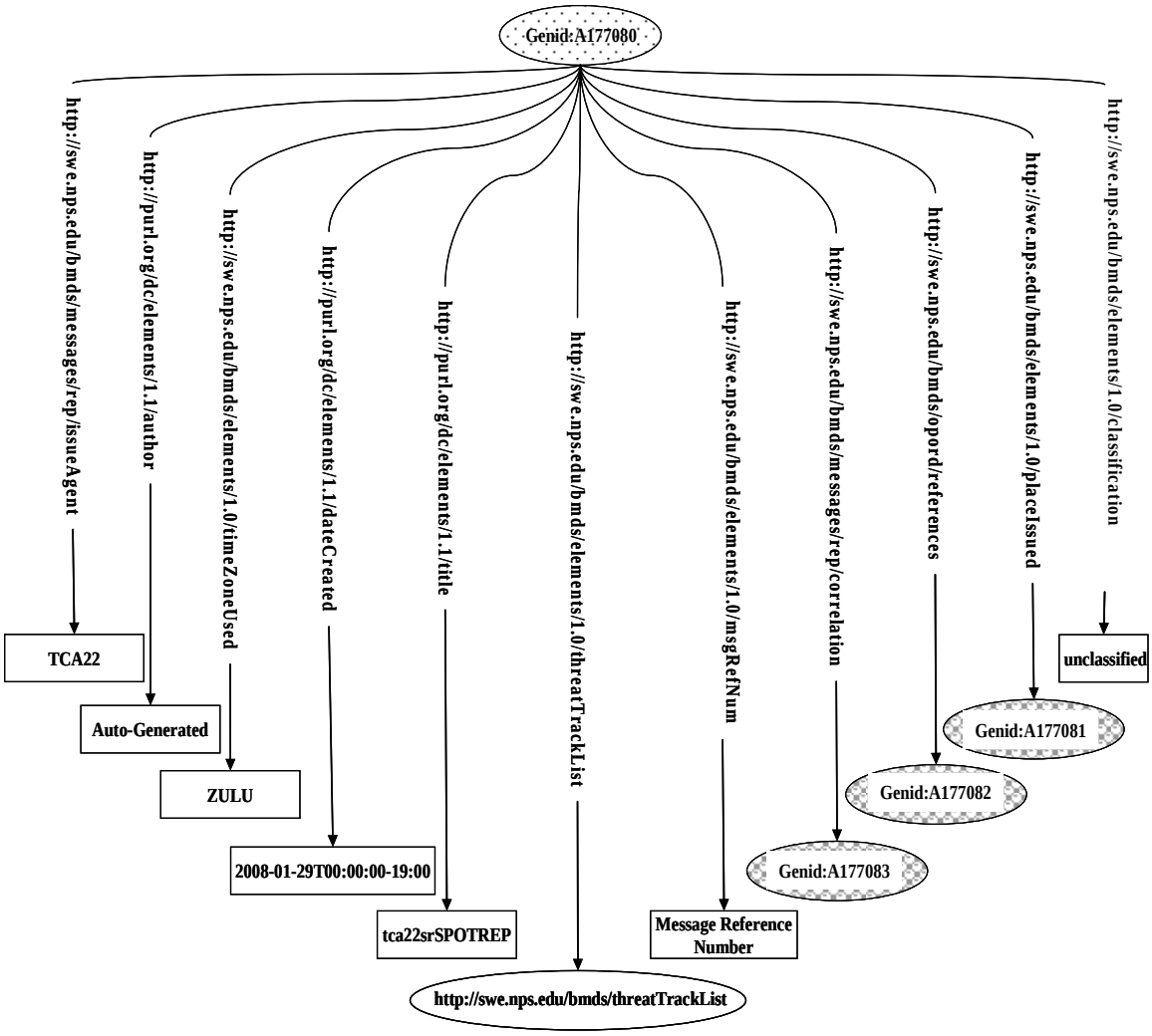

Figure 32. SPOTREP RDF Graph Biography Information

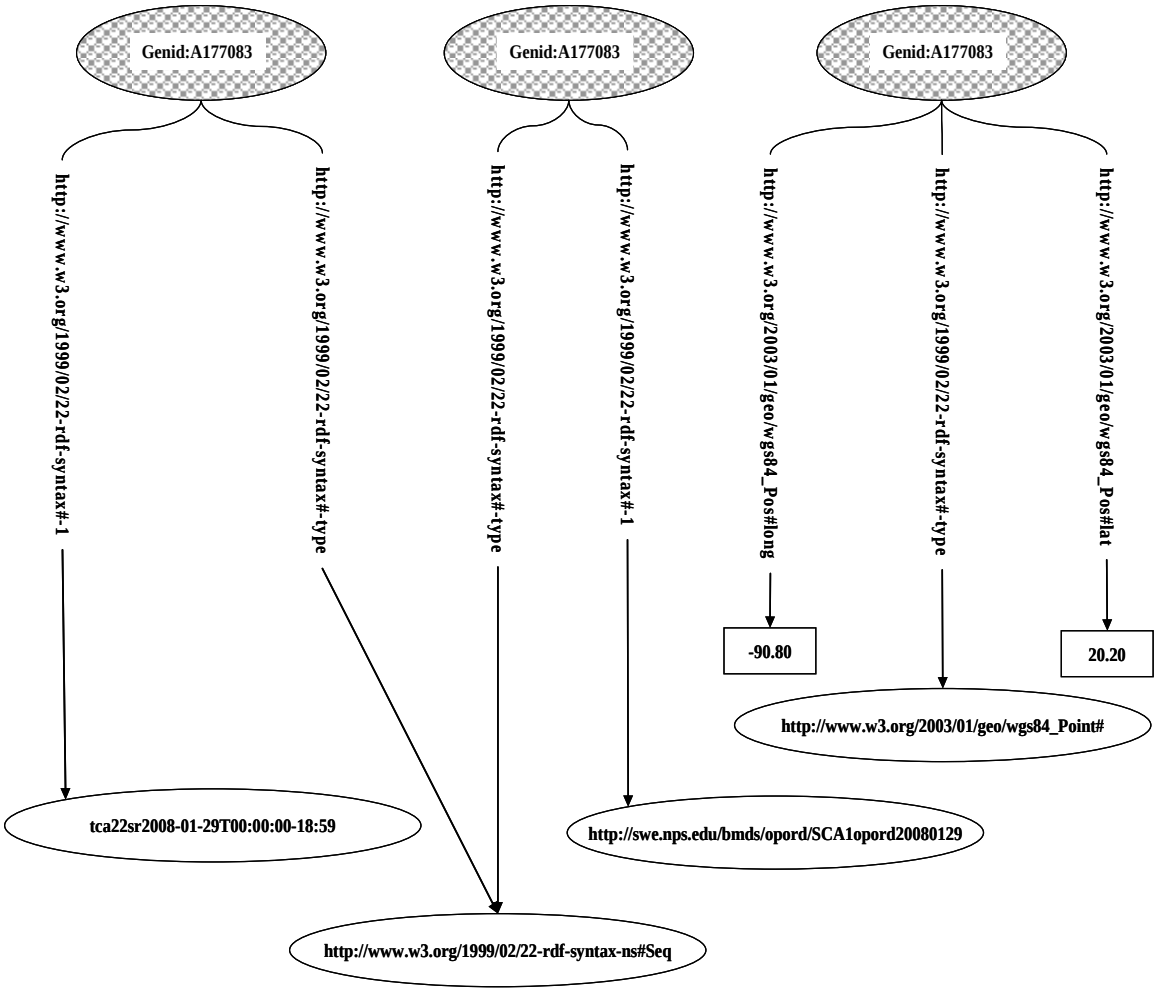

Figure 33. SPOTREP RDF Graph

C. CONCLUSION

Machine-processable messages such as the SPOTREP defined and discussed above are essential in providing the feedback necessary to self-regulate the collection of agents that are collectively tasked to achieve the objectives of a battle. The SPOTREP message in these scenarios provides an automated means to update the kill-chain services performance with respect to the MOE and MOO from the bottom of the chain of command to the top of the chain of command.

Given the original objective of the battle given by the OPORD and the SPOTREPs that come from various authorized entities to an agent is all the collective

knowledge arriving at an agent (i.e., TCA, RCA and SCA) contain the information that creates the situational awareness of that agent about the ongoing battle. In order to generate and disseminate the subsequent commands that make the battle progress towards the objective stated in the OPORD, the agent must consider the following:

1. The rules of engagement in-force for the given battle
2. Obligations and prohibitions in effect for the particular agent as specified in the agent's duties.
3. The strategies permitted to use in conducting a battle

These can be automated using a set of rules that take some human input and some automated rules in a rule execution engine such as RuleML [44] that has been specially derived to be used in the Semantic Web. A non sematic-web related version of this form was presented in constructing the SCAs, RCAs and TCAs in [29]. An accurate modeling of this intelligence would require security, the probabilistic nature of observations, and the distributed execution of rules in the orchestration that we leave as future work.

VI. CONCLUSION

A. CONTRIBUTIONS

In this research, we have demonstrated that by introducing the contributions listed below time-critical C4ISR applications can be realized.

1. Proposed Extensions to WSDL Standard to Encompass MOEs and MOPs

The proposal for extending the WSDL standard to encompass MOPs and MOEs provides the necessary support to conduct dynamic compositions. MOPs and MOEs are significant parameters used to measure the performance and effectiveness of systems and services with respect to assigned missions at the tactical, regional, and strategic levels of the military command structure.

2. Incorporated MOEs and MOPs to Support Service Selection and Performance Feedback

Binding MOPs and MOEs directly to the executable code provides two significant advantages to the orchestration of services in a time-critical SOA-based system. The first advantage is, in addition to advertising what function a service performs the system can now advertise how well the service performs its function and how effective the service is. The second advantage is that upon completion of a service, execution monitors can record the MOPs and MOEs and update the WSDL, thus providing a means of building up a historical basis-of-confidence for each service giving potential clients as much information as possible for selecting the appropriate service for use in their processes.

3. Tailored Machine Readable OPORD

The machine processable OPORD is a significant and crucial piece of this research as it is the element that kicks off the entire C2BM process. By representing the OPORD in RDF, the OPORD can be automatically distributed, manipulated, or otherwise processed, as either an entire document or portion of a document.

We tailored the standard template for an OPORD to support inclusion of MOPs and MOEs. In addition, our approach provides for automatically monitored services at runtime to record MOP and MOE data until the service completes execution at which time the data is converted into statistics that are in turn used to update the WSDLs.

The machine readable OPORD relieves operators from having to conduct previously laborious and error-prone tasks such as sifting through the OPORD to find needed information to create new OPORDs, rewriting tasks to subordinate units, and delivering the OPORD to subordinate commands; all of these tasks can be automated using web-based services with the machine-processable OPORD.

4. Shadow Pattern, Run-time Monitoring and Performance Feedback

Applying the shadow pattern [32] in BPEL orchestrations provides a high level of assurance that the orchestration will complete its assigned process in its allotted time. By invoking local and remote services concurrently, the local service can serve as the shadow to the remote service. The remote services might provide some higher-level quality of service than the local service. However, should the remote service not return a result within a prespecified time the BPEL alarm construct would execute the next step of the process using the result provided by the local service.

Run-time monitoring serves two purposes. One purpose is to provide a means within BPEL to measure time-constrained tasks. At the beginning of an orchestration a local time service is started and upon orchestration completion the local time service is stopped. The other purpose is to monitor both the MOP and MOE of the related task. In this research, we monitored the percentage of defended assets that survived a missile attack and rolled the results up the chain-of-command to produce the MOO.

Lastly, our approach provides for a performance feedback mechanism. The monitoring service once complete, records the information for the results for the QoS, MOPs, MOEs, and MOOs. This information is passed to a distribution service that forwards the results to the developers for inclusion into the services basis-of-confidence, which are updated and then exposed through the WSDL.

B. FUTURE WORK

A significant amount of research remains to be done before time-critical C4ISR SOA-based systems and applications are to be deployed into the battlespace. Chief among the research areas for further study are real-time networks, operating systems, and security up and down the SOA protocol stack. In addition to real-time and security challenges there remains significant work to be done in identifying and optimizing algorithms for the selection of services for orchestration at compile-time. Further research is then necessary to consider the dynamic selection of services at run-time, including locating, binding, and orchestrating services on-the-fly. This directly exposes the need for verification and validation of services for use in such orchestrations. Finally, some near-term research that directly extends this dissertation is the addition of probabilistic information to represent the uncertainty of the MOPs and MOEs.

1. Security

C4ISR applications in the missile defense and other military domains have demanding security requirements. This combined with the security considerations for Web services creates a different and challenging environment in which to build trusted systems. That which makes Web services so enticing, such as exposing standardized interfaces to new and existing applications, loose coupling, spanning organizational networks, technology agnosticism, and dynamically re-configurable processes, also makes Web services to some extent disagreeable from a security perspective. We highlight three significant security areas that require research to overcome considerable challenges. From this dissertations perspective the three most significant areas considered are multi-level security, application and network security, and coalition security.

a. Mult- Level Security(MLS)

The ability to both continuously and correctly execute the kill chain in a timely manner is of paramount importance in the BMDS. The BMDS is a distributed system of systems that processes sensitive information in an environment where not all users have the same clearance, need to know. In addition, the systems comprising the

BMDS may not even have the same set of security policies in place. In the BMDS and in the larger C4ISR domain, information must flow between such systems and users without violating security policies. The DoD defines multilevel security in [40] as

> …a capability that allows information with different sensitivities (i.e., classification and compartments) to be simultaneously stored and processed in an information system with users having different security clearances, authorizations, and needs to know, while preventing users from accessing information for which they are not cleared, do not have authorization, or do not have the need to know. MLS capabilities often can help overcome the operational constraints imposed by system-high operations and can foster more effective operations. For example, systems once separated by an *airgap* or connected only by a *sneaker net* may be electronically interconnected by an *MLS guard*, allowing the data transferred to be current rather than merely historical in value.

Research in MLS has been going on for years within DoD, but MLS in a SOA environment creates new challenges. Interestingly enough there is at least one effort that is addressing the challenges of introducing MLS of C4ISR systems into a SOA environment, in particular for the GIG, and that is the Joint Cross Domain eXchange JCDX [41]. JCDX is data labeling technology used in the DoD's Horizontal Fusion program designed to permit access to data across multiple security domains. The original JCDX design supported only highly specialized applications by small numbers of users. The JCDX program realized that with DoD moving to the GIG network architecture "systems would no longer require the use of dedicated clients but instead would be accessible to any client over the network." and began engineering a Web-services solution for its product, as illustrated in Figure 34.

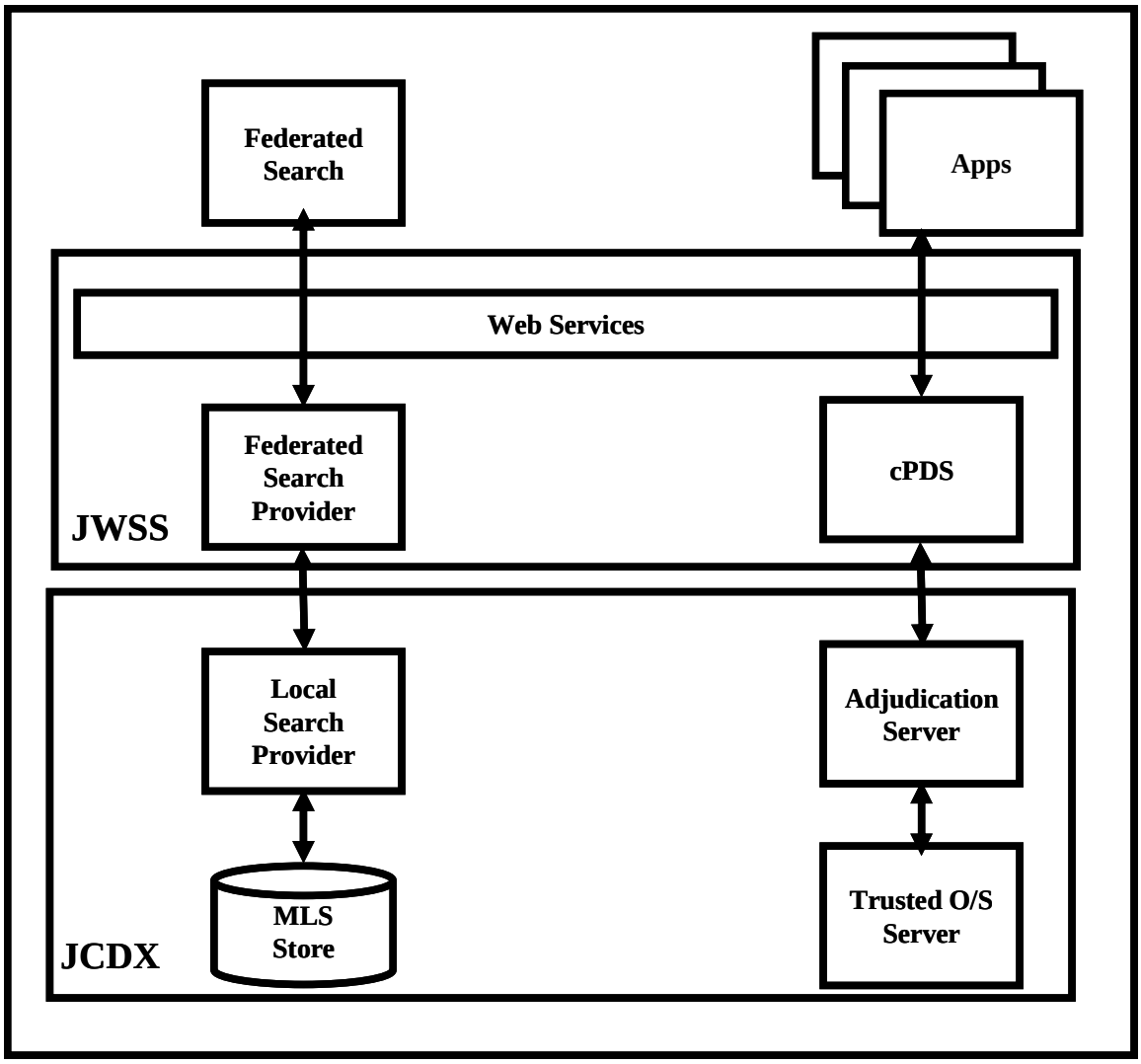

Figure 34. JCDX Web Services Architecture

The JCDX currently provides two Web services. The first is the classification Policy Decision Service (cPDS). The cPDS' primary purpose is to "provide other systems with methods for handling labeled data such as label comparison [41]." The second service is the Federated Search Provider (FSP). The FSP "allows users and applications to search multi-level data stores from single level networks and provides a 'read down' capability to all lower level domains."

However, as [41] points out, there is still significant research to be done before full MLS can be offered. [41] addresses the consumption of content and does not

address issues associated with the production of content such as label production from the source. There must also be trust established between systems, which in current non-SOA deployments are statically defined. Accreditation is another significant challenge identified in [41]. Some open-ended questions include

- "How do we put all of these individually trusted services together to produce a single trusted service?"
- "Does the composition of many trusted services yield a single trusted service, or do we need some sort of trusted path?"
- "Is trusted path a valid concept in SOA?"
- "Should we look at trusted transactions, to include the state of the transaction and the data in the transaction?"

These are only a few of the issues and challenges that require attention, but there has been some work on MLS for SOA.

b. Network

In a system of systems it is unacceptable to have any piece of the network compromised: The result can be worse than a hard real-time requirement being missed, such as if a message in the kill chain is unknowingly modified to redirect an interceptor launch so that the intercept misses its target. [42] discusses multiple needs and the importance of securing Web services in the business domain. We address these same concerns, but with respect to processes within the C4ISR domain.

The network is of significant concern especially in a SOA environment that executes time-critical C4ISR applications. The traditional distributed computing security of isolating systems and users on networks and sub-networks is not conducive to a SOA: This actually defeats one of the primary advantages of using a SOA as it limits access to potential service producers and consumers.

Firewalls are the first line of defense in the network, but will only work in the Web services if appropriate policies are in place and executed in a firewall architecture that encompasses packet filtering, circuit and applications, and stateful inspection.

Intrusion detection should play a role in the network for a SOA based C4ISR application. As [42] points out an "IDS solutions raise concerns that an attack may be taking place," but in the C4ISR domain this simply is not enough. "What is needed is a more proactive approach that determines the susceptibility to attacks before networks are compromised which is provided by vulnerability assessment."

The C4ISR warfighting domain has stringent network communication requirements. Cryptographic functions can enable the application security requirements of authentication, confidentiality, message integrity, and non-repudiation, but it is still an open question as to how to determine the appropriate set of security policy within this domain.

At the Application layer traditional security protocols such as Secure Sockets Layer (SSL), Transport Layer Security (TLS), and Internet Protocol Security (IPSec) are available to offer encryption and authentication. The issue as [42] suggests is that while WS type SOA offers integration across organizational boundaries there must be a similar offering of security integration across those same boundaries and "without this security integration, security solutions remain at a per project level, with no central means of configuring, monitoring, analyzing, and controlling integration data flows." At the heart of this is the XML security standards that exist today, but it is imperative that we begin implementations of these standards to identify shortfalls.

Most of the challenges identified here have some degree of work done within standards groups to help fill the security needs for Web services, but we need to start implementing standards such as WS-Security and WS-Policy to see how well they meet the requirements of time-critical C4ISR applications. In addition, there is a significant amount of work to do within the C4ISR domain specifically in the warfighter sub-domain to determine the exact requirements and policies.

c. Coalition

Within the DoD, interoperating with allied and coalition partners is a requirement. We must overcome issues related to disclosure and releasibility. While JCDX addresses some of the problem, it does not address it in its entirety. The entire process for deciding what is releasable should be automated.

All of the challenges addressed as research topics above are moving forward and making varying degrees of progress. Another issue to consider is time. If we cannot execute the security policies and complete all security tasks within guaranteed repeatable cycles, we cannot effectively accomplish many of the warfighter missions; that is, there is a balancing act to be be performed in considering the tradeoff between security effectiveness and operational effectiveness.

2. Real-time Requirements

As discussed previously C4ISR applications in the warfighter domain have hard real-time requirements. For the SOA to work in this application domain, orchestrations, and services must operate within a hard real-time run-time environment. There is ongoing research this area. The vision for such a run-time environment would equate to Web services running on real-time Web application servers across a real-time network (service bus). In the SOA environment, the QoS must be guaranteed from end-to-end of the distributed process.

Some significant work currently taking place is under Sun Microsystems' Java Community Process, JSR-50: The Distributed Real-Time Specification for Java, with the aim of specifing:

> Distributable Real-Time Threads
>
> This is a proven programming model for constructing sequential control flow applications with end-to-end timeliness properties in distributed systems. The DRTSJ's distributable threads are supported by a "real-time" implementation of Java's Remote Method Invocation, as originally proposed in JSR-50;

Distributable Thread Integrity Framework

This framework allows application designers to plug appropriate policies for maintaining the health and integrity of distributable threads in the presence of failures; and

Scheduling Framework

This framework allows application designers to plug appropriate user space policies for scheduling distributable and local threads.

The effort is significant in that it leverages some of the more successful efforts in this research area such as RT-CORBA 1.0 and 2.0 and Sun Microsystems' recent reference implementation of JSR-1: Real-Time Specification for Java (RTSJ).

3. Dynamic Selection of Services at Run-Time

Our research considers orchestrating kill chains prior to deploying them for use on Web servers—ready orchestrated kill-chain services. A significant contribution to this work would be to design an approach that dynamically orchestrates a kill chain at run-time; selecting and updating services as the kill chain is executing. This would in essence identify the most appropriate atomic level services to accomplish each executing task to provide the best possible solution for the executing process.

4. Algorithms for Service Selection

Identifying the best possible service from a set of competing alternative services requires detailed analysis of both the process that is executing and the Quality of Service offered by the individual services. Research into identifying the best possible approach is necessary. In our research, we considered only exemplar QoS, MOP, MOE, and MOO. There are literally hundreds of these types of parameters and simply identifying appropriate ones for this domain is a significant undertaking. Further, identifying algorithms to mix and match services based on these parameters for the best possible orchestration solutions is even more of an undertaking. Opportunities for Operations Research and Systems Analysis (ORSA) research in this area abound.

5. Statistical Representation of MOEs and MOPs

MOPs and MOEs are a significant contribution in this research, but we show them as numbers: They represent results that summarize statistical computations performed over observed data. A significant research effort would be to extend the MOP and MOE definitions to express the latter details quantifying the significance (or lack thereof) in observed MOO, MOP, MOE and QoS values. Such a representation offers a deeper semantical representation to the clients that are inclined to further combine such statistical results as they go up the chain of command. For instance we could add to the MOP and MOE definitions to the distribution from which the sample (random variable) comes from, add the confidence level, and other appropriate statistics. Information such as this provides a potential consumer with invaluable information with which can be taken into account when making binary decisions out of statistical feedback from the operational environment.

LIST OF REFERENCES

[1] Vice Admiral Arthur K. Cebrowski, U.S. Navy, and John J. Garstka, "Network-Centric Warfare: Its Origin and Future," U.S. *Naval Institute Proceedings*, January 1998 http://www.usni.org/Proceedings/Articles98/PROcebrowski.htm., January 2008.

[2] D. S. Alberts, J. J. Garstka, and F. P. Stein, Network Centric Warfare; Developing and Leveraging Information Superiority, CCRP publication series, 6th printing, April 2005.

[3] Director, Force Transformation, Office of the Secretary of Defense, The Implementation of Network-Centric Warfare,www.oft.osd.mil, January 5, 2005.

[4] Deputy Secretary of Defense, Office of Primary Responsibility, Assistant Secretary of Defense for Networks and Information Integration, DoD Directive 8100.1, "Global Information Grid (GIG) Overarching Policy," September 19, 2002, http://www.dtic.mil/whs/directives/corres/html/81001.htm., June 2007.

[5] The Defense Information Systems Agency, The NECC Provisional Technical Transition Architecture Specification, Version 0.5.7. April 12, 2006, http://www.ditco.disa.mil/News/Documents/File/NECC_ptta_v0_5_7.pdf. June 2007.

[6] G. Gardner, J. Baummstrack, P. Sego, "Industry Best Practices in Achieving Service Oriented Architecture (SOA)", *A report of the Net-Centric Operations Industry Forum* (NCOIF), Data Sharing and Sevices Startegy Working Group, Association for Enterprise Integration (AFEI), Arlington, VA, April 2005.

[7] K. Birman, R. Hillman, S. Pleisch, Building Net-Centric Military Applications over Service Oriented Architectures, April 2007 http://www.cs.cornell.edu/projects/quicksilver/public_pdfs/GIGonWS_final.pdf.

[8] No author given, FCS Overview, September 2005, http://www.army.mil/fcs/factfiles/overview.html., October 2007.

[9] D.S. Caffall, "Developing Dependable Software for A System-of-Systems", Ph.D., United States Naval Postgraduate School, Monterey, CA, March 2005.

[10] The Defense Information Systems Agency, "NCES SOA Foundation Working Group Charter", DRAFT, March 16, 2006.

[11] Missile Defense Agency, "A day in the life of the BMDS", BMDS booklet, third edition, http://www.mda.mil/mdalink/pdf/bmdsbook.pdf., June 2006.

[12] Lt. Gen, C.E. Croom Jr., "Service-Oriented Architectures in Net-Centric Operations". Crosstalk, The Journal of Defense Software Engineering, Software Technology Support Center, Hill Air Force Base, Ogden, UT., pp. 13-15, July 2006.

[13] S. Haddad, P. Moreaux, S. Rampacek, "Client Synthesis For Web Services By Way Of A Timed Semantics", presented at ICEIS 2006, May 23-27, 2006.

[14] E. Christenson, F. Curbera, G. Meredith, S. Weerawarana, "Web Services Description Language 1.1", http://www.w3.org/TR/2001/NOTE-wsdl-20010315, © 2001. January 2006.

[15] A. Alves, et al., "Web Services Business Process Execution Language Version 2.0", Public Review Draft, August 23, 2006.

[16] T. Bellwood, et.al., "UDDI Version 3.0.2, UDDI Spec Technical Committee" Draft, Dated October 19, 2004.

[17] M. Gudgin, Et.al., "SOAP Version 1.2 Part 1: Messaging Framework", *W3C Recommendation* June 24, 2003 http://www.w3.org/TR/soap12-part1., January 2006.

[18] M. Gudgin, Et.al., "SOAP Version 1.2 Part 2" *W3C Recommendation* 24 June 2003, http://www.w3.org/TR/soap12-part2/., Accesssed February 2006.

[19] H. Haas, Et. al., "SOAP Version 1.2 Specification Assertions and Test Collection", *W3C Recommendation* 24 June 2003, http://www.w3.org/TR/soap12-testcollection. January 2006.

[20] The Defense Information Systems Agency, "Introduction to Service-Oriented Architecture Foundation (SOAF)", http://www.disa.mil/nces/product_lines/soa.html, February 2007.

[21] T. Erl, "Service-Oriented Architecture", *Concepts, Technology, and Design*, Prentice Hall Technical Reference, Upper Saddle River, NJ, 2005.

[22] Wikepedia, SOA definition, http://en.wikipedia.org/wiki/Service-oriented_architecture, March 2007.

[23] OASIS, SOA definition, http://www.oasis-open.org/committees/tc_cat.php?cat=soa, March 2007.

[24] Headquarters, Department of The Army, (2005), *Field Manual 5-0, Army Planning and Orders Production*, Headquarters, Department of The Army, Washington, D.C., http://www.army.mil/usapa/doctrine/Active_FM.html, January 17, 2007.

[25] S. Powers, “Practical RDF”, O’Reilly & Associates, Inc., Sebastopol, CA, 2003.

[26] R. Paul, J. Srivastava, and D. Wijesekera, “Information Quality Based (Computer/Distributed) System Evaluation”, in International Journal of Testing and Evaluation, pp. 41-52, June 2000.

[27] Department of The Army, “Transformation Focus - Situational Awareness *SOA in The DOD”*, http://www.army.mil/armybtkc/focus/sa/soa-dod.htm, September 2007.

[28] K. Birman, R. Hillman, S. Pleisch. Building Net-Centric Military Applications over Service Oriented Architectures. SPIE Defense and Security Symposium 2005. March 29-31, 2005. Orlando, FL.

[29] D.Wijesekera, J. B. Michael, and A. Nerode, “BMD Agents: An Agent-Based Framework to Model Ballistic Missile Defense Strategies”, I*n Proc. 6th Int. Workshop on Policies for Distributed Systems and Networks*, IEEE, Stockholm, Sweden, pp. 115-118, June 2005.

[30] U.S. Department of Defense. Department of Defense Dictionary of Military and *Associated Terms*. Joint Pub. 1-02, April 12, 2001 (as amended through May 23, 2003).

[31] A. D’Ambbrogio, “A Model-driven WSDL Extension for Describing the QoS of Web Services”, International Conference on Web Services (ICWS’06), Chicago, USA, September 18-22, 2006, 0-7695-2669-1/06, IEEE, Computer Society.

[32] T. W. Otani, M. Auguston, T. S. Cook, D. Drusinsky, J. B. Michael, and M. Shing, "A design pattern for using non-developmental items in real-time Java", Proceedings of the 5th international workshop on Java technologies for real-time and embedded systems (JTRES 2007), Vienna, Austria, September 26 - 28, 2007, pp. 135-143.

[33] D. Beckett, “RDF/XML Syntax Specification (Revised)” *W3C Recommendation* February 10, 2004, http://www.w3.org/TR/rdf-syntax-grammar/. September 2007.

[34] A. Alves, et al., Business Process Execution Language, OASIS Standard, April 11, 2007.

[35] Gen. T Hobbins, “Cultural Shift”, C4ISR Journal August 1, 2007, http://integrator.hanscom.af.mil/2007/August/08302007/08302007-22.htm. Accessed November 2007. April 2008.

[36] D. Zimmerman, “J.E.B. Stuart: Battle of Gettysburg Scapegoat”, Published Monday, June 12, 2006 in America's Civil War, http://www.historynet.com/jeb-stuart-battle-of-gettysburg-scapegoat.htm., June 2007.

[37] D.B. Weller, D.C. Boger, and J. B. Michael. Command structure of the Ballistic Missile Defense System. In M.J. Savoie, H.W. Chu, J. Michael, and P. Pace, editors, *Proc. Int. Conf. on Computing, Communications and Control Technologies*, pp. 42-48, Austin, TX., August 2004. Int. Inst. of Informatics and Systemics.

[38] S. Graham,D. Davis, S. Simeonov, G. Daniels, P. Brittenham,Y. Nakamura, P. Fremantle, D. Konig, and C. Zentner, "Building Web Services with Java, Making sense of XML, SOAP, WSDL, and UDDI", Second Edition., Sams Publishing, Indianapolis, IN, 2005.

[39] J. B. Matjaz, "Business Process Execution Language for Web Services," Second Ed., 2006, Packt Publishing Ltd., Birmingham, UK.

[40] No Author, Multilevel Security in the Department Of Defense: The Basics., 1 March 1995. at http://nsi.org/Library/Compsec/sec1.html, April 20, 2007.

[41] C. J. Raney, "Integrating Multilevel Command and Control into a Service Oriented Architecture to Provide Cross Doamin Capability," CCRTS, *The State of the Art and the State of the Practice*., SPAWAR, Systems Center, San Diego, CA, 2006.

[42] M. P. Papazoglou, "Web Services: Principles and Technology," Pearson Education Limited, 2008.

[43] Headquarters, Department of the Army, *FM 6-99.2U.S., ARMY REPORT AND MESSAGE FORMATS*, Headquarters, Department of the Army,Washington, DC, 30 April 2007, http://www.army.mil/usapa/doctrine/Active_FM.html, January 17, 2007.

[44] No Author given, RuleML, The Rule Markup Initiative, http://www.ruleml.org/, March 15, 2008.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor James Bret Michael
   Department of Computer Science
   Naval Postgraduate School

4. Professor Man-Tak Shing
   Department of Computer Science
   Naval Postgraduate School
   Monterey, California

5. Professor Dan C. Boger
   Department of Computer Science
   Naval Postgraduate School
   Monterey, California

6. Professor Doron Drusinsky
   Department of Computer Science
   Naval Postgraduate School
   Monterey, California

7. Professor Duminda Wijesekera
   Department of Computer Science
   George Mason University
   Fairfax, Virginia

8. Dr. Butch Caffall
   NASA IV&V Facility
   Fairmont, West Virginia