

ONTOLOGY-BASED DATA FUSION WITHIN A NET-CENTRIC  
INFORMATION EXCHANGE FRAMEWORK

by

Ho Jun Lee

---

Copyright © Ho Jun Lee 2009

A Dissertation Submitted to the Faculty of the  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2009

THE UNIVERSITY OF ARIZONA  
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by **Ho Jun Lee** entitled **Ontology-based Data Fusion within a Net-centric Information Exchange Framework** and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy in Electrical and Computer Engineering.

\_\_\_\_\_ Date: August 06, 09  
Bernard P. Zeigler, Ph.D.

\_\_\_\_\_ Date: August 06, 09  
Jonathan Sprinkle, Ph.D.

\_\_\_\_\_ Date: August 06, 09  
Roman Lysecky, Ph.D.

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

\_\_\_\_\_ Date: August 06, 09  
Dissertation Director: Bernard P. Zeigler, Ph.D.

## STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder

SIGNED: Ho Jun Lee

## ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor, Professor Bernard P. Zeigler, who has the attitude and the substance of a genius: he continually and convincingly conveyed an inspiration and encouragement from the preliminary to the concluding level which enabled me to develop an understanding of the subject. Without his guidance and persistent help this dissertation would not have been possible.

I would like to thank my committee members, Professor Jonathan Sprinkle and Professor Roman Lysecky for their encouraging words and efforts to improve this dissertation during busy semesters.

I would like to thank to Professor Nathan Goodman and Professor Young Jun Son who devoted their time and energy to teach and help me.

I would like to show my special thank to Dr. Doohwan Kim, who has supported me.

In addition, I thank Chungman and Lahiru who encouraged me and shared the same enthusiasm for the work.

This dissertation is dedicated with my love to Eun Mi, Eunchae and Seoyeon, and to all my family including my parents, my brother and my law-in-parents for their love, trust, and support during this long education.

## TABLE OF CONTENTS

LIST OF ILLUSTRATIONS.....	9
LIST OF TABLES.....	12
ABSTRACT .....	13
CHAPTER 1. INTRODUCTION .....	15
1.1 Motivation and Contributions .....	15
1.2 Organization of the Dissertation .....	21
CHAPTER 2. BACKGROUND RESEARCH .....	22
2.1 System Entity Structure (SES) Ontology .....	22
2.1.1 SES ontology concept .....	22
2.1.2 Ontological operations in SES .....	25
2.2 Information Exchange Framework (IEF) .....	27
2.3 Data Fusion (DF) Process by Information Exchange Framework (IEF) .....	29
2.3.1 DF process in Information Exchange Framework (IEF) .....	29
2.3.2 Joint Directors of Laboratories (JDL) DF model .....	29
2.4 Distributed Data Fusion System in Service Oriented Architecture (SOA) .....	31
2.4.1 Data Fusion (DF) architecture .....	31
2.4.2 Service Oriented Architecture (SOA) .....	32
2.4.3 Distributed fusion architecture in SOA concept .....	33

## TABLE OF CONTENTS – Continued

2.5 Review for Multi-level Information Fusion Techniques .....	35
2.5.1 Level 2 and Level 3 (high-level) fusion processes .....	35
2.5.1.1 Situation Awareness (SA) .....	35
2.5.2 Level 1 fusion process .....	37
2.5.2.1 Kalman Filter (KF) and Nearest Neighbor (NN) with gating association technique for multi-sensor tracking .....	38
2.5.2.2 Cursor-on-Target (CoT) to alleviate interoperability problem between sensors for Level 1 fusion .....	41
2.6 Battle Management Language (BML) .....	45
2.6.1 BML grammar .....	47
<b>CHAPTER 3. DATA FUSION PROCESS IN SES INFORMATION EXCHANGE FRAMEWORK .....</b>	<b>51</b>
3.1 SES Ontology Implementation for Data Fusion Process.....	51
3.2 BML for Pragmatic Frame .....	55
3.2.1 Basic BML implementation in SES ontology .....	56
3.2.2 Revision of BML paragraph for pragmatic frame .....	58
3.3 High-level Information and Pragmatic Frame .....	61
3.3.1 More SESs for SA of high-level information .....	62
3.3.2 A Multi-level SES ontological DF process in IEF .....	67

## TABLE OF CONTENTS – Continued

3.4 Proof-of-Concept Examples .....	69
3.4.1 Example 1.....	69
3.4.2 Example 2 .....	71
<b>CHAPTER 4. AUXILLIARY ISSUES FOR LEVEL 1 DATA FUSION</b>	
<b>PROCESS .....</b>	<b>74</b>
4.1 Multi-sensor Tracking with NN and Gating .....	74
4.2 Improve Interoperability between Radars with Cursor-on-Target (CoT) .....	78
<b>CHAPTER 5. EVALUATION IN WARGAME SCENARIOS .....</b>	
5.1 Scenario 1-1 .....	81
5.1.1 Scenario description .....	81
5.1.2 DEVS models .....	82
5.1.2.1 DEVS formalism .....	83
5.1.3 The results of simulation under scenario 1-1 .....	92
5.2 Scenario 1-2 .....	93
5.2.1 Scenario description .....	93
5.2.2 The results of simulation under scenario 1-2 .....	95
5.3 Scenario 2 .....	96
5.3.1 Scenario description .....	96
5.3.2 DEVS models.....	99
5.3.3 The results of simulation under scenario 2 .....	102

## TABLE OF CONTENTS - Continued

CHAPTER 6. CONCLUSIONS AND FUTURE WORK .....	104
6.1 Conclusions .....	104
6.2 Future Work .....	105
APPENDIX I NATURAL LANGUAGE STYLE DESCRIPTION OF RADAR-SES .....	107
APPENDIX II RADAR-SES SCHEMA .....	109
APPENDIX III PRUNED RADAR-SES IN XML DOCUMENT FORMAT .....	116
APPENDIX IV BML-SES SCHEMA .....	118
APPENDIX V COT 2.0-SES SCHEMA .....	129
APPENDIX VI COT 3.0-SES SCHEMA .....	131
APPENDIX VII COT 3.0 XML FILE WITH RADAR DATA .....	137
APPENDIX VIII DATUM CONVERSION FOR TRACKDISPLAY .....	140
REFERENCES .....	144

## LIST OF ILLUSTRATIONS

Figure 1-1 Overview of the ontological Data Fusion (DF) system .....	19
Figure 2-1 Basic SES representation .....	23
Figure 2-2 Representation of a book in SES .....	24
Figure 2-3 A pruned Book-SES .....	26
Figure 2-4 SES structure components .....	26
Figure 2-5 Generic Information Exchange Framework (IEF) .....	28
Figure 2-6 JDL process model .....	30
Figure 2-7 Data Fusion (DF) architecture .....	32
Figure 2-8 Generic webservice architecture .....	33
Figure 2-9 One iteration of Kalman Filter (KF) algorithm .....	39
Figure 2-10 Illustration of the gating technique .....	41
Figure 2-11 CoT structure and relationships among elements .....	45
Figure 2-12 Diagram to illustrate the role of BML .....	46
Figure 3-1 Natural language style description in SES builder .....	52
Figure 3-2 SES tree representaion of radar data .....	53
Figure 3-3 A pruned Radar-SES .....	54
Figure 3-4 BML as a pragmatic frame .....	56
Figure 3-5 Basic BML representation in SES (not revised version) .....	57
Figure 3-6 A pruned BML with an order.....	58

## LIST OF ILLUSTRATIONS – Continued

Figure 3-7 Extended SES for BML .....	60
Figure 3-8 UserTargetRelation-SES .....	62
Figure 3-9 Illustration of relative direction .....	64
Figure 3-10 Relationships between relative directions .....	65
Figure 3-11 Threat-SES .....	66
Figure 3-12 Overview of SES IEF via BML .....	69
Figure 3-13 Pruned BML-SES for Level 1 .....	71
Figure 3-14 Pruned BML-SES for Level 3 .....	73
Figure 4-1 Two cases of ambiguous association .....	74
Figure 4-2 MathML representation of a matrix.....	77
Figure 4-3 CoT 2.0-SES .....	79
Figure 4-4 CoT 3.0-SES .....	80
Figure 5-1 Illustration of scenario 1-1 .....	82
Figure 5-2 A scenario generator .....	85
Figure 5-3 DEVSJAVA input console for scenario specification .....	85
Figure 5-4 XML document output for scenario 1-1 .....	86
Figure 5-5 DEVSJAVA models in Simview .....	87
Figure 5-6 State diagram of <b>commander</b> model .....	90
Figure 5-7 Snapshot of TrackDisplay .....	91

## LIST OF ILLUSTRATIONS - Continued

Figure 5-8 XML description of scenario 1-2 .....	93
Figure 5-9 Illustration of scenario 1-2 .....	94
Figure 5-10 XML description of scenario 1-2 .....	97
Figure 5-11 Illustration of scenario 2 .....	98
Figure 5-12 DEVSJAVA models for scenario 2 in Simview .....	99
Figure 5-13 State diagram of <b>unit</b> model .....	100
Figure 5-14 State diagram of <b>commander</b> model .....	101
Figure 6-1 Information Processing Efficiency (IPE) .....	106
Figure A-7-1 Relations between ECEF and LLA .....	141

## LIST OF TABLES

Table 2-1 CoT 2.0 main schema .....	42
Table 2-2 CoT 3.0 beta schema .....	44
Table 5-1 Simulation specification for scenario 1-1 .....	81
Table 5-2 Changes of relations and threat types in scenario 1-1 .....	92
Table 5-3 Simulation specification for scenario 1-2 .....	93
Table 5-4 Changes of relations and threat types in scenario 1-2 .....	95
Table 5-5 Simulation specification for scenario 2 .....	97
Table 5-6 Changes of relations and threat types in scenario 2 .....	102

## ABSTRACT

With the advent of Network-Centric Warfare (NCW) concepts, Command and Control (C2) Systems need efficient methods for communicating between heterogeneous systems. To extract or exchange various levels of information within the networks requires interoperability between human and machine as well as between machine and machine. This dissertation explores the Information Exchange Framework (IEF) concept of distributed data fusion sensor networks in Network-centric environments. It is used to synthesize integrative battlefield pictures by combining the Battle Management Language (BML) and System Entity Structure (SES) ontology framework for C2 systems. The SES is an ontology framework that can facilitate information exchange in a network environment. From the perspective of the SES framework, BML serves to express pragmatic frames, since it can specify the information desired by a consumer in an unambiguous way. This thesis formulates information exchange in the SES ontology via BML and defines novel pruning and transformation processes of the SES to extract and fuse data into higher level representations. This supports the interoperability between human users and other sensor systems. The efficacy of such data fusion and exchange is illustrated with several battlefield scenario examples.

A second intercommunication issue between sensor systems is how to ensure efficient and effective message passing. This is studied by using Cursor-on-Target (CoT), an effort to standardize a battlefield data exchange format. CoT regulates only a few essential data

types as standard and has a simple and efficient structure to hold a wide range of message formats used in dissimilar military enterprises. This thesis adopts the common message type into radar sensor networks to manage the target tracking problem in distributed sensor networks. Multi-sensor target tracking is one of the main issues in the tracking community. A Web Service concept is applied to the multiple sensor environment expressed in CoT message format.

To demonstrate the effectiveness of the proposed Information Exchange Framework for data fusion systems, we illustrate the approach in an air defense operation scenario using DEVS modeling and simulation. The examples depict basic air defense operation procedure. The demonstration shows that the information requested by a commander is delivered in the right way at the right time so that it can support agile decision making against threats.

## CHAPTER 1. INTRODUCTION

### 1.1. Motivation and Contributions

Outline of orders for military Operations generally consists of five sections in the following order: Situation, Mission, Execution, Service Support, and Command and Signal [1]. Commanders take the first step along with understanding the current situation and future threats in planning military operations. Therefore, gathering valuable information from diverse and accurate sources is necessary and urgent work for commanders to comprehend the situation. Furthermore, the more refined information is considered as more valuable information for decision-making process of C2 systems because it gives more intuitive information and shortens the reaction time for commanders to decide the course of action. Network Centric Warfare (NCW) is a new doctrinal concept of warfare in the Information Ages to fight and win victory against challenges by the potential power of information superiority connecting all available military assets [2]. Commanders can achieve information advantage with NCW. Networking, including Webservices, offers opportunities as well as challenges for C2 systems. NCW raises two main issues. The first issue is to study networking technologies such as architectures or routing algorithms. On the other hand, which is our main interest in this dissertation, is to develop methodologies to achieve information superiority from various information sources. This issue can be divided into two sub-research areas: how to combine information from diverse sensors and produce valuable information is the one, and how to interact with information sources to extract the valuable information is

another. The former is about Data Fusion (DF) process in the net-centric environment. The latter concerns Information Exchange Framework (IEF) in the networks. As a result, although NCW seeks increased data availability, it requires also a new paradigm for C2 systems to use it in network-centric environment including a strategy to integrate C2 systems with DF systems in an Information Exchange Framework.

The study of a new approach to connect C2 systems with DF systems begins with examination of DF system from the perspective of the SES ontology concept. Since we have multiple sources of information, how to integrate all available information has received interest from the military community. Data Fusion (DF) is a process to refine knowledge from various information sources and bring about integrated pictures of the battlefield [3][4]. Hence, this issue is closely related to implementation of the NCW doctrine to provide valuable information to C2 systems. There are two fundamental architectures of DF systems [6][7]. For radar networks, the central scheme is more popular. However, more networked environments need a more distributed strategy of fusion [6]. DF systems produce disparate levels of information in the JDL model [3][5][6]. Although all levels of information are important to commanders, a study of SES ontology-based Situation Awareness (SA) for high-level DF process enables DF process to connect into SES ontology framework. Moreover, we need effective ways to interact with the distributed fusion architecture in net-centric environments. This needs to establish a second connection between C2 systems and DF systems. By introducing BML

as a pragmatic frame of Information Exchange Framework, we can express commanders' requests and the response of DF systems in an effective way.

In general, DF systems collect all data in a central fashion and broadcast basic and key information to C2 systems. C2 systems then re-process it with respect to itself to generate customized information by human or computers if it is required. This architecture does not support interaction between C2 systems and DF systems. Our integration scheme can give a systematic method to provide customized information to C2 systems. Some recent research in [58][59] focuses on the user roles in DF process by suggesting additional fusion level, called user refinement level in the Joint Directors of Laboratories (JDL) model. Situation Awareness, based on an ontology concept, is investigated in [37][38]. Since their ontologies were defined in OWL, we need to extend their approach based on SES ontology for C2 systems. System Entity Structure (SES) was invented to represent structures of system models for modeling and simulation [8], and it is extended as a simulation-based data model approach in the network environment [9][10][11][12]. The SES Ontology organizes information in a hierarchical manner. It places its attention on the roles of information users in the information exchange process in networks and constructs a framework to communicate information called Information Exchange Framework (IEF). It gives a way to exchange data messages by tailoring their structures according to requirements specified in a pragmatic frame. This pruning process reduces communication traffic since pruning minimizes the information volume. References [10][11][12] investigate the SES pruning process in network traffic analysis and weather

service. We extended this concept to the DF process in sensor networks in [13]. C2 systems need a message format which is inter-communicative between different systems; the message format formulates the C2 systems' requirements as well. C2 systems usually are involved with humans. On the other hand, DF systems are automated machinery. An attempt for automated message exchange between C2 systems and simulated forces is Battle Management Language (BML) [14]-[22]. BML is being developed to increase interoperability between real C2 systems and simulated troop operations. The main objective of BML is to fill in the gap between human language, more specifically used by military people, and machine understandable language through defining an intermediate language which can be understood by both sides. BML is a well formalized language and part of the multinational operational language called Coalition BML (C-BML) [13][14]. Some efforts to apply BML are discussed in [23][24]. BML is capable of expressing the user's requirements in an explicit way, and because of this it can be exploited as a tool to give form to pragmatics [13].

In addition, we think about another interoperability issue between dissimilar machines for sensor level fusion. Very often, different sensors use different data types for their observation data. To overcome this diversity, we need to use an interoperable message format, which can be understood among various types of sensors. This need leads another study about Cursor-on-Target (CoT) [25][26]. CoT is a structured machine-to-machine message format for passing key targeting data. It concentrates on What, Where, When (WWW) information, which is used to share the same situational picture or target

information among systems. It has a 2.0 version, which is stable and has successfully implemented several applications [25][27][28]. Currently, the CoT 3.0 version has been developed for multi-target information passing in [27].

The first contribution of this dissertation is to design a conceptual architecture in which DF systems are integrated with C2 systems in SES Information Exchange Framework (IEF). The design concept of this Ontological Data Fusion (ODF) architecture is shown in Figure 1-1.

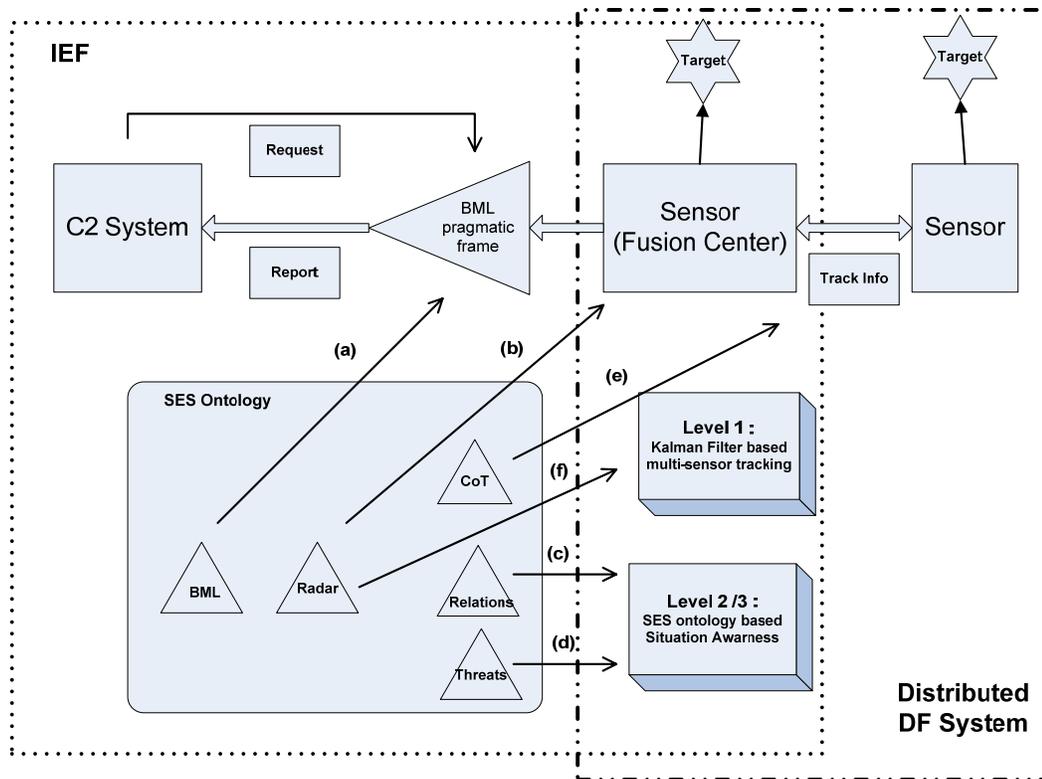


Figure 1-1 Overview of the ontological Data Fusion (DF) system.

First, we develop several SESs of BML, Radar, Relations, Threats, and CoT. The BML-SES is an extended version to formulate pragmatic frames by expressing a commander's

requests for information and sensor reports as indicated by arrow (a) in Figure 1.1. The Radar-SES is an ontological representation of radar data, as shown in (b). The Relations-SES and Threats-SES are intended to define relation and threat types for ontology based SA for Level 2 and Level 3 information of the JDL model as shown in (c) and (d). We develop pruning and transformation operations for ontological Situational Awareness (SA). CoT is adopted to support sensor-to-sensor message passing employing an SES ontology formulation as shown in (e). We develop a restructuring operation to attach Radar-SES under CoT. The target's temporal and spatial data are used to perform multi-sensor tracking for the Level 1 process in the JDL model as shown in (f). We apply the Kalman Filter (KF) and Nearest Neighborhood (NN) with the gating association technique to produce fused track data in a distributed environment. The approach casts the data fusion process development within an ontological framework that is amenable to modeling and simulation.

The second contribution is that this view can facilitate the autonomous information exchange between humans and machines as well as machines and machines by investigating two interoperability issues of human-to-machine as well as machine-to-machine via BML and CoT. The human's intervention can calibrate the process, and we can obtain customized information. Therefore, successful communication between human and machine is one of the main topics in the DF process. Moreover, each sensor understands other data types in multisensor environments to share the information

between systems in an automated fashion. As a result, the interoperability between machines enables the ODF system's automation with minimum human intervention.

## 1.2. Organization of the Dissertation

We describe background knowledge including the SES ontology concept, JDL DF model with Kalman Filter and Situation Awareness, and Battle Management Language and Cursor-on-Target in Chapter 2. Chapter 3 addresses the formulation of IEF for ODF. In addition, the extension of BML to represent a pragmatic frame for multi-level data fusion process is discussed as well. For Level 1 we describe the NN and the gating technique, and CoT message format in Chapter 4. Two examples show the roles of the pragmatic frame by BML. Simulations under war-game scenarios explain the effectiveness of proposed ODF in IEF. The dissertation's summary and future work are presented in Chapter 5.

## CHAPTER 2. BACKGROUND RESEARCH

### 2.1. System Entity Structure (SES) Ontology

#### 2.1.1. SES ontology concept

Ontology is a study concerned with the nature of existence of things and their relationships [9][29][30][31]. It contains classes (elements), attributes of the classes, and relationships between classes with which to represent or model knowledge of a certain domain. System Entity Structure (SES) is a formal ontology framework to represent the elements of a system (or world) and their relationships in hierarchical manner [9]. It provides a model to describe knowledge of a domain in a structural way. Since it is originated from the representation of the model structure, SES is easily accommodated in modeling and simulation for automation. While SES represents complex data in a rigorous way, it has flexibility and efficiency to change the structure according to a variety of choices. Figure 2-1 shows the basic representation elements of the SES.

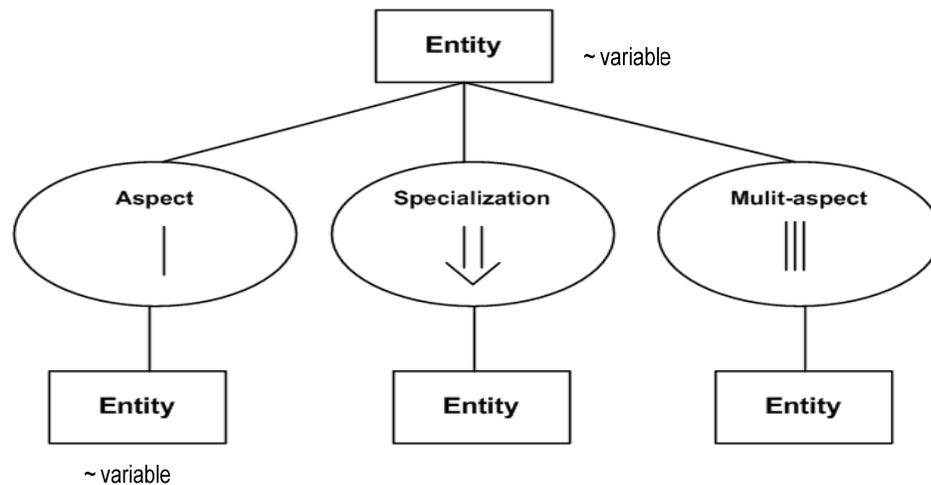


Figure 2-1 Basic SES representation [9]

SES consists of entities, (multi-)aspects, specialization, and variables.

- Entities represent things that have existence in a certain domain. Entities can have variables which can be assigned a value within given range and types
- Aspects represent ways of taking things apart into more detailed ones and labeled decomposition relation between the parent and the children.
- Multi-aspects are aspects for which the components are all of the same kind.
- Specialization categorizes things in specific forms that it can assume. It is a labeled relation that expresses alternative choices that a system entity can take on.
- Entities can have variables, which can be assigned a value within given range.

Moreover, six axioms are presented in [9]:

- Uniformity: Any two nodes that have the same label have identical variable types and isomorphic sub-structures.

- Strict hierarchy: No label appears more than once down any path of the tree structure.
- Alternating mode: Each node has a mode which is one of the following: entity, aspect, or specialization. If one node is entity, the successors of the entity are aspect or specialization. If the node is aspect or specialization, its children are entity. The root must be entity.
- Valid brothers: No two brothers have the same label.
- Attached variables: No two variable types attached to the same item have the same name.
- Inheritance: The parent and any child of a specialization combine their individual variables, aspects and specializations when pruning is activated.

For example, a book can be represented in SES structure in Figure 2-2.

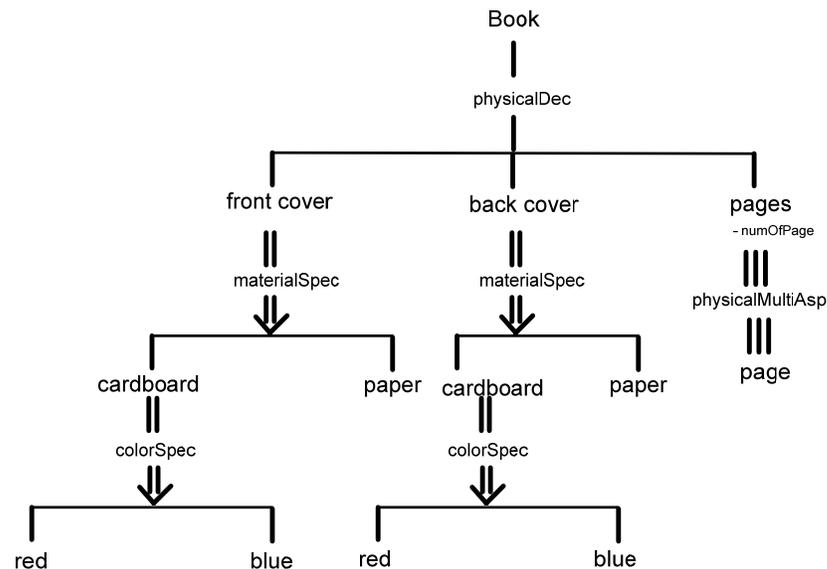


Figure 2-2 Representation of book in SES

A book consists of front cover, back cover, and pages, which show the physical decomposition relation between book and covers. The front cover of a book can be made of either cardboard or paper. The cardboard is also manufactured in red or blue. Pages contain multiple entities of the same characteristics. Pages has a variable of numOfPage in integer format.

#### 2.1.2. Ontological operations in SES

The SES operations causing structural change to extract specific information are: pruning, restructuring, and transforming [9]. Pruning is an operation to cut off unnecessary structure in a SES based on the specification of a pragmatic frame. More specifically, it includes processes: a) to assign particular values to variables of entities, b) to trim the SES and get the minimal SES for end-users by picking specific elements from multiple choices. Restructuring is a mapping process within the same domain and may result in the alternative structures. Transforming is also a mapping process, but from one domain to another domain.

One choice of a book pruned by a user's demand from book-SES is shown in Figure 2-3.

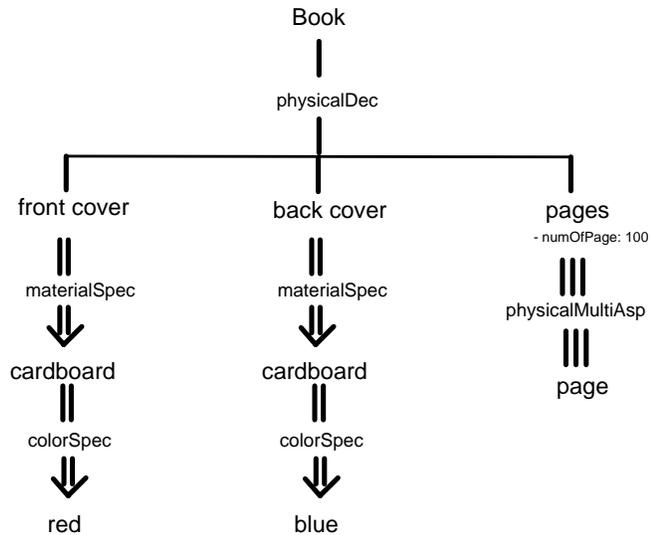


Figure 2-3 A pruned book-SES

The pruned book-SES means that the book has a front cover and a back cover, both with red cardboard and 100 pages of volume.

A set-theoretic specification of SES structure is suggested in Figure 2-4.

*SES = < Entities, Aspects,  
 Specializations, rootEntity,  
 entityHasAspect, entityHasMultiAspect,  
 entityHasSpecialization, aspectHasEntity,  
 MultiAspectHasEntity, specializationHasEntity,  
 entityHasVariable, MutiAspectHasVariable,  
 VariableHasRange >*

Figure 2-4 SES structure components

The pruning process reduces selections of *entityHasAspect*, *specializationHasEntity*, *MultiAspectHasEntity* relations above. After completing pruning, there should be no choice left in the above relations. Moreover, at the implementation level, a pragmatic frame is able to choose anything in the ontology. For example, an information client might request to be continually updated on a one entity variable value, like the current time. Then that pragmatic frame results in a simple sub SES structure with one end-entity.

A SES can be represented in XML format. XML is an appropriate markup language for SES representation, since it can easily add user-defined tags, which can describe them without any restriction [32]. It is natural to represent hierarchical structure as well. More details are addressed in Chapter 3.

## 2.2. Information Exchange Framework (IEF)

Such ontological operations are invoked by the users' requirements in Information Exchange Framework [9]. Since the user's requirements specify the structural change of SES, we emphasize the roles of users or information consumers in information exchange scheme, called pragmatic frame.

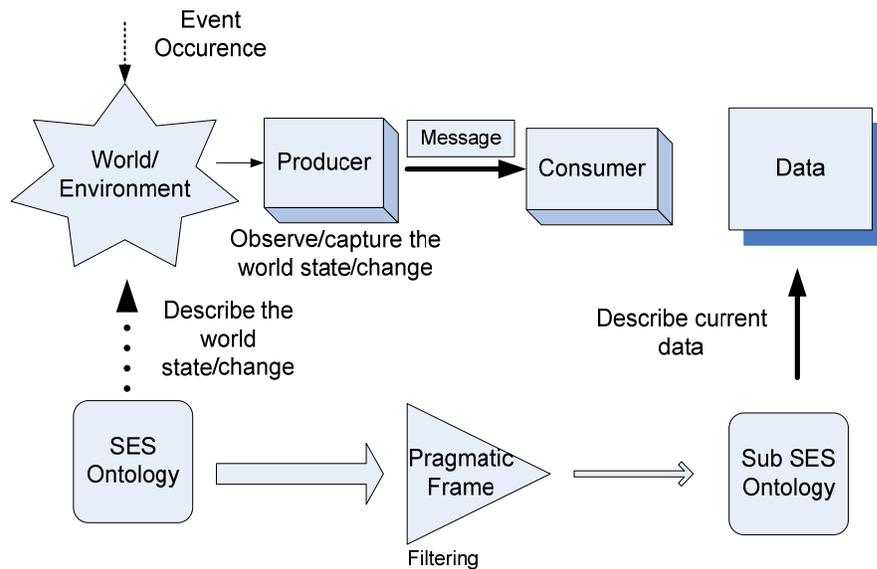


Figure 2-5 Generic Information Exchange Framework (IEF)

The general procedure of information exchange is shown in Figure 2-5. A producer generates and provides the information. A user or consumer needs some information and requests the information, which may cause alteration of the ontology. In SES concept, a producer designs a master SES ontology, which represents the available information of a domain, and a consumer wants to know specific information, which is contained or implied in the master SES structure. The producers are information providers. They capture data and turn them into meaningful information according to the users' demand. That requirement, formalized as a pragmatic frame, can lead to some processing on the SES that results in a sub-SES, which is tuned to the consumer's requirements [9].

## 2.3. Data Fusion (DF) process by Information Exchange Framework (IEF)

### 2.3.1. DF process in Information Exchange Framework

A Data Fusion (DF) combines data from multiple sensors or sources in order to improve interpretation of these data [3]. Data fusion, or information fusion, process uses techniques to integrate similar or diverse data for more refined detection, tracking, classification, situation awareness, and threat assessment [4]. Networking large numbers of military data sources brings up technical issues on how to combine all information or data for common and shared battle field pictures, which is equivalent to a data fusion process. Since IEF is a systematic concept of a way to refine raw data by pragmatic frame, the IEF concept is closely related to DF processes in networks. We explore a subject of DF process in IEF from the perspective of the SES ontology.

### 2.3.2. Joint Directors of Laboratories (JDL) DF model

Several process models such as JDL, Waterfall, and Omnibus have been proposed. JDL is a well-known DF processing model for applications to military domains [3][5]. It defines several functional levels from 0 to 4.

- Level 0 is a source pre-processing step on the sensor level. We are interested in how to associate and characterize signals to do higher level processes.
- Level 1, Objective Refinement, concerns refining the representation of individual objects. For example, we fuse multiple sensor data to track multiple targets in areas of interest on this level.

- Level 2, Situation Refinement, describes the current relationships among objects and events. Objects are clustered, and we aggregate situations and object groups.
- Level 3, Threat Refinement, projects the current situation into the future to support choices among alternative courses of action as consequences of level 2's results.
- Level 4, Processing Refinement, concerns monitoring and controlling other processes to optimize the knowledge created by other processes.

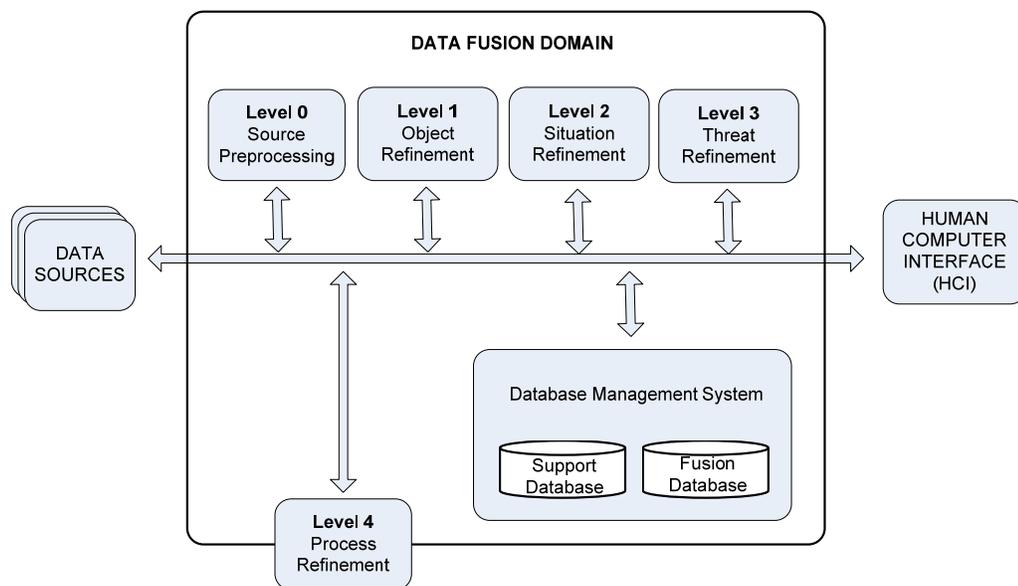


Figure 2-6 JDL process model [3]

Figure 2-6 shows the DF structure and data flow between different DF levels in the JDL model.

We focus on the Level 2, and partially Level 3 for high-level fusion processes in Chapter 3. High-level fusion is a study of relationships among objects and events of interest within a dynamic environment in an abstract manner [5]. It is an analysis process of data

obtained by sensors. Moreover, we think of IEF as Level 4 process since IEF regulates Level 1, Level 2, and Level 3 process according to user's requirements, and it extracts exact information in an effective way by reducing unnecessary data. Level 1 issue is also studied as an auxiliary issue in Chapter 4.

## 2.4. Distributed Data Fusion System in Service Oriented Architecture (SOA)

### 2.4.1. Data Fusion (DF) architecture

The architecture of a DF system is generally divided into two categories: centralized and distributed fusion architectures [6] (also called Multi-sensor Data Fusion (MSDF) in [3]) and Distributed Sensor Network (DSN) in [3]). Central fusion architecture presumes there is a central fusion center which collects all possible raw data or tracks and fuses them in a central manner and the results are distributed to various users. Another approach assumes there is a network composed of multiple sensor nodes. The first variation is that these nodes are arranged in a hierarchy, and low level nodes dealing with sensor data send the processing result to higher level nodes to be fused. On the other hand, in a fully distributed manner, there are no hierarchical relations between nodes. Each node can communicate with others and acts as a fusion center. So they track their own targets and send the data to others via communication channels. The central fashion is optimal and simple if we have fully large enough communication channels to send all the sensor data without communication delay, and we equip high speed and large computational

resources. The distributed fashion has some advantages, including less need of processing power, lower communication load, and faster response to the user demands [7]. Therefore, the attention of this dissertation is paid to distributed architectures in network environments with the objective of improving Level 1, 2, and 3 process performances.

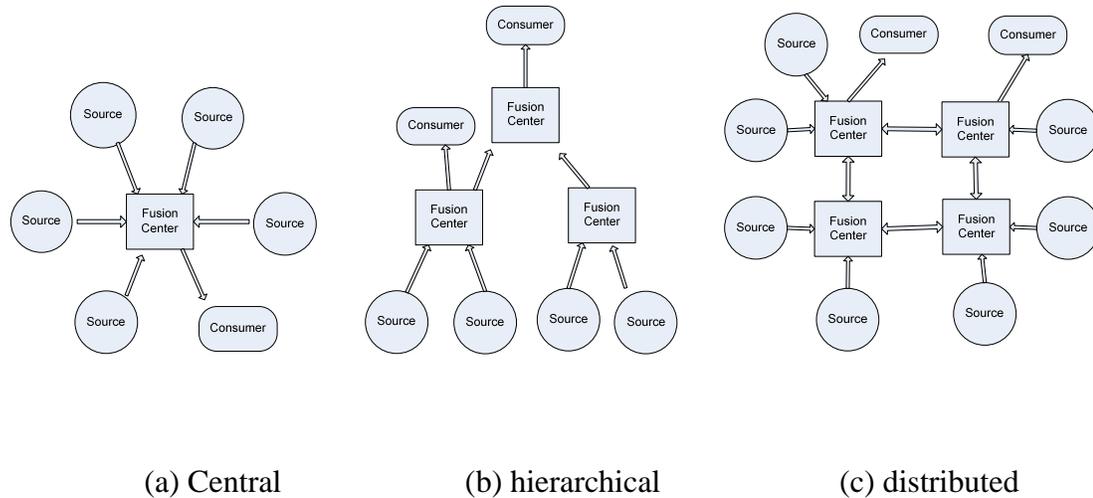


Figure 2-7 Data Fusion (DF) architectures [6]

#### 2.4.2. Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) is a methodology with which a new application is created through integrating existing and independent processes, which are distributed over networks. SOA considers a message as an important unit of communication so it can be regarded as a “message-oriented” service. Web services architecture [33][34] is an implementation of SOA, which is based on exchanging messages, describing web services, and publishing and discovering web service descriptions. Web services are described by Web Services Description Language (WSDL) [35], which is XML-based language providing the required information, such as message types, signatures of

services, and a location of services, for clients to consume the services. Publishing and discovering web service descriptions are managed by Universal Description Discover and Integration (UDDI) [36] which is a platform-independent and XML style registry.

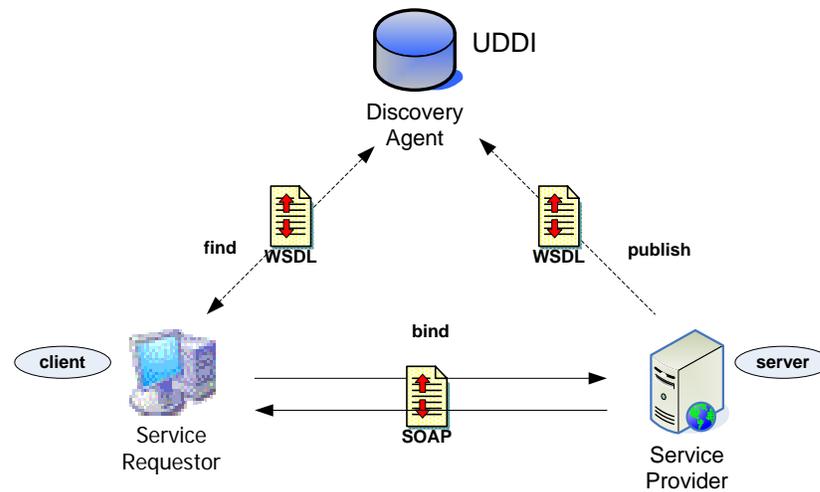


Figure 2-8 Generic Web Service architecture [33]

#### 2.4.3. Distributed fusion architecture in SOA concept

The Net-centric Warfare (NCW) concept seeks to connect all possible resources and forces for more competitive and efficient war-fighting advantages. All nodes of the network can communicate with each other and have certain functions that can affect other nodes. Each sensor can be a fusion center and play as a node of network, which means it has a fusion function like tracking of targets and transferring the local tracks to other nodes, which, in turn, has its local tracks and can carry out a fusion function like track-to-track correlation to form shared common pictures of battlefield for C2.

The design concept of the data exchange architecture in the distributed multi-sensor network we propose here is based on architecture (c) in Figure 2-7. Each distributed fusion center with sources covers a part of the battle field and maintains the local air-pictures by communicating with other fusion nodes. The center provides services relating to the battle area pictures to users in a manner akin to that of web service producers. If the node is not available for the request, then it returns information of an available fusion center to the requester according to register profile, which contains others' information availability, just as the discovery agent in web services does. Therefore, a local fusion center is not only a service provider but it is also a service broker.

The dynamic situation is understandably more complex than the static case. When a consumer or a provider changes its spatial position with respect to time propagation, the information availability of the provider gets a different state so that each fusion center has to update its register profile with other fusion centers' reports on a regular basis. In fact, web service concept is more efficient and powerful in dynamic situations since we are easily able to add or remove services.

Another advantage of this SOA concept is that it can improve a higher level of data fusion process, such as level 2 or level 3, along with ontological description of the information. Levels 2 and level 3 are more related to information of the user's circumstances. A threat can be critical to a location of some units, but not to others. The SES information exchange framework is another enabler to achieving high-level fusions.

## 2.5. Review for Multi-level Information Fusion Techniques

### 2.5.1. Level 2 and 3 (high-level) fusion processes

This section introduces several techniques to produce various levels of information in the DF process. For high-level information, various techniques are investigated [5]. However, ontology-based Situation Awareness (SA) gets newly interested in the DF community. We address the basic concept of ontological SA.

#### 2.5.1.1. Situation Awareness (SA)

Situation Awareness (SA), or Situation Assessment, is a study to recognize the relations between entities (objects) and the situations of circumstances based on the relations. Endsley's mention about SA is a popular concept of it; "Situation Awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." [30] It is, therefore, a study about Level 2 and for partially related with Level 3 based on results of Level 1 for high-level fusion process [5]. Level 1 process clarifies entity identification and characterization which increases the knowledge of entities in the area and time of interest. With the knowledge of entities we find out all relevant relations among entities. Finally we project the entities and relations to the near future to predict influence of entities. The main objective of SA is to provide supports for operators' (referred as users or customers in this paper) need [31]. Hence it is important to coordinate with operators' interest, which is considered as pragmatic frame in IEF. Ontology-based SA is recently stimulating research interest in various communities. A

few previous studies have explored this issue in [37]-[41]. The authors define relations and situation ontology in OWL [42]. Then they recognize specific relations between entities, which relations, in turn, describe the current situation according to pre-defined rules of the relations, which is described in RuleML [43].

The SA is a process of figuring out relations of objects. A relation is a subset of the Cartesian product of a number of sets. The Cartesian product is a subset of pairs of elements of A and B [30]:

$$A \times B = \{ \langle a, b \rangle ; a \in A, b \in B \} \quad (\text{Equation 2-1})$$

Then relation  $R$  is a subset of the Cartesian product.

$$R \subset A \times B \quad (\text{Equation 2-2})$$

In logic, relation is a semantic concept corresponding to the predicate. Therefore, relations mean interpretation of predicates. Inference, or reasoning, is a process of drawing conclusions by applying inference rules to either the axioms of a given theory or to previously drawn new theorems. The axioms or theorems are addressed in terms of predicates. Since the predicates can be interpreted as relations, the inference process draws true statements about relations.

Several challenges to implement computer-based SA dictated are discussed in [30].

- **Situation as Objects:** Situations or relations have to be treated as objects to recognize and handle situations or relations on computers. We treat this problem by applying the SES ontology implementation scheme.
- **Relation Derivation Algorithm:** SES ontological operations, like pruning and transformation, are applied as a derivation algorithm.
- **Relevance of Relations:** Since we choose my application domain as military air defense operations, we can restrict our scope and define some relations which can be drawn from Level 1 target information.
- **Complexity of Derivation Algorithm:** We do not use a complex search algorithm to match the relations with rules. We simply compare them in If clauses.

An automation inference process can be implemented by modeling a matching process between a set of relations (found from knowledge of objects) and a set of pre-defined relation rules to draw conclusions (or situations). This dissertation follows a similar logical reasoning or inference process in SES ontology for automatic SA for air defense operations.

### 2.5.2. Level 1 fusion process

Since high-level information is generated from lower level information, we need to consider some techniques which can facilitate getting Level 1 information in network environment. We consider the Kalman Filter (KF) based multi-sensor tracking algorithm and CoT, which is a message format to promote message exchange between sensors.

### 2.5.2.1. Kalman Filter (KF) and Nearest Neighbor (NN) with gating association technique for multi-sensor tracking

This part is involved in the Level 1 process of the JDL model. It is a process to reduce uncertainty about object properties, including location and velocity. The Kalman Filter (KF) is a popular recursive estimation technique, which is accepted as a useful and efficient estimator in a variety of areas [5][44][45][46]. Estimation aims at obtaining an estimated state with minimum errors from measured data. It is based on a state-based formulation of continuous or discrete systems even though the following is for discrete systems. There are some assumptions for KF derivation: Linear and time-variant systems and the state is a random variable under Gaussian assumption. We omit detailed derivation and address the whole iterative process.

The transition and measurement system modeling are shown in Equation

$$X(k) = F(k-1)X(k-1) + G(k-1)U(k-1) + w(k) \quad (\text{Equation 2-3})$$

$$Y(k) = H(k)X(k) + v(k) \quad (\text{Equation 2-4})$$

where  $k = 1, 2, 3, \dots$  which is a time index,  $X(k)$  is a state vector at time  $k$ ,  $F(k-1)$  is a state transition matrix from previous time to current time,  $U(k-1)$  is a random input, and  $w(k)$  is a noise term compensating for a state transition modeling error, which is driven from  $N[0; \sigma_w]$  which represents Normal distribution with zero mean and  $\sigma_w$  of standard deviation.  $Y(k)$  is a measurement given at current time  $k$ ,  $H(k)$  is a

measurement matrix,  $v(k)$  is an observation noise, which is driven from  $N[0; \sigma_v]$ ,  $[\cdot]^T$  denotes transpose operation of a matrix, and  $[\cdot]^{-1}$  denotes inverse operation of a matrix.

Then one iteration cycle is shown in Figure 2-9.

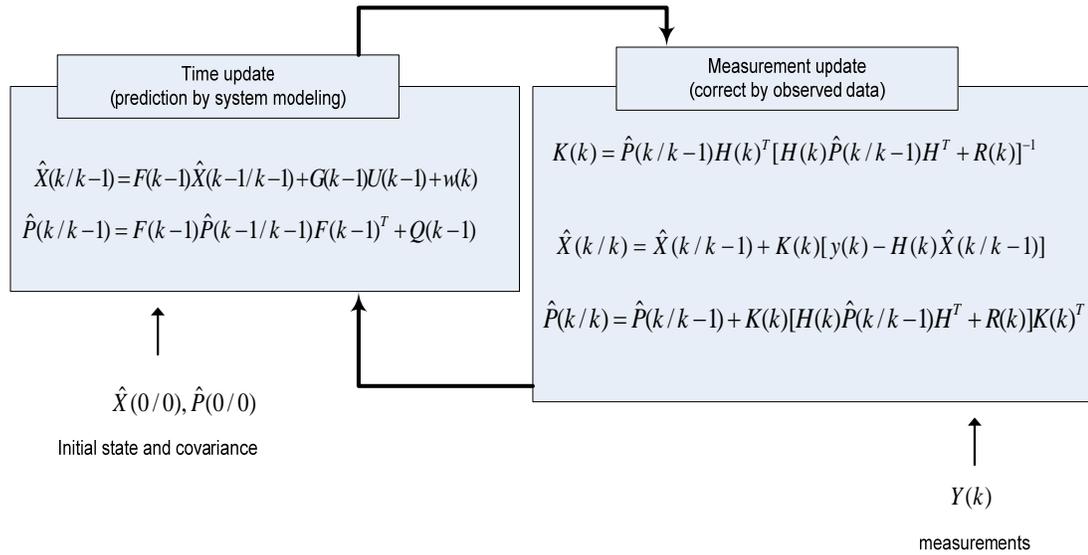


Figure 2-9 One iteration of Kalman Filter (KF) algorithm

where  $\hat{X}(k/k)$  is a state estimate vector, and  $\hat{P}(k/k)$  is a state covariance matrix, which means the quality of estimated state,  $K(k)$  is Kalman gain,  $Q(k-1)$  is obtained by  $E[w(k)w(k)^T]$ , and  $R(k)$  is  $E[v(k)v(k)^T]$ .  $E[\cdot]$  represents expectation value of a random variable. We assume that  $v(k)$  and  $w(k)$  are Gaussian random variables.

One iteration consists of two general steps: time update and measurement update. Time update is carried out according to the modeling of system dynamics based on the previous

state estimate. It is a step to predict the current state based on state propagation models. Measurement update is accomplished by means of observed data through sensors. It compensates for the time update error with the Kalman gain. Kalman gain means the portion of new information of the state obtained from measurements. Therefore, Kalman gain plays a role of adjusting the predicted state error with new information in current data.

Now we extend our interest to the multi-sensor problems. When we deal with tracking problems in multiple sources, an association issue comes up. Even without a false alarm or clutters, ambiguity happens between targets and measurements of each sensor. Existing research has targeted this issue, and we can find some efficient techniques in [47][48][49].

In this dissertation we adopt a simple gating technique and Nearest Neighborhood (NN) for target tracking in multiple sensors when some ambiguity takes place during estimation process because this topic is the interest of the dissertation. Gating is originally a mathematical technique to reduce the computational complexity of association between measurements and targets.

A measurement validation gate is established by

$$[Y(k) - \hat{Y}(k)][H(k)\hat{P}(k/k-1)H^T + R(k)]^{-1}[Y(k) - \hat{Y}(k)]^T \leq \kappa \quad (\text{Equation 2-5})$$

where  $\kappa$  is found out in a table of chi-square distribution. The graphical representation of gate is given in Figure 2-10.

If a measurement falls into the gate, which means it satisfies the Equation 2-5, we can assume that the measurement is a candidate for updating the state. If more than one candidate measurement exists in the volume of the ellipsoid, we choose a measurement with minimum Euclidian distance (Nearest Neighborhood). We extend it to solve the uncertain association between targets of different sensors.

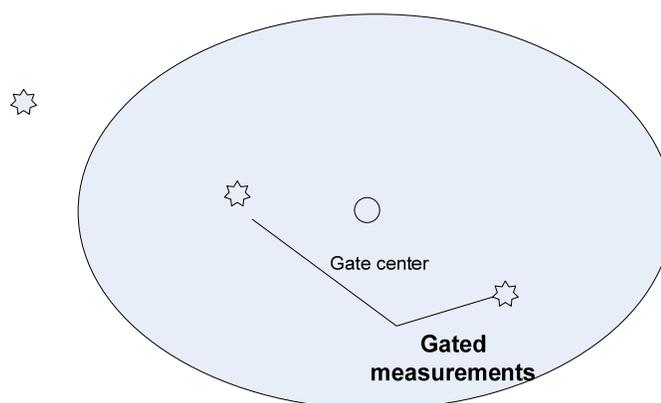


Figure 2-10 Illustration of the gating technique

#### 2.5.2.2. Cursor-on-Target (CoT) to alleviate interoperability problem between sensors for Level 1 fusion

In a nutshell, CoT is a XML-based message type to represent a data structure. It is devised to exchange key information of What, When, and Where (WWW) between different systems. The terminology was inspired by a comment said by Air Force Chief of Staff General John Jumper. He mentioned the need for net centric operational capabilities

via machine-to-machine interoperability by saying; “the sum of all wisdom is a cursor over the target” [25][26].

Traditional combats have required humans to transfer time-sensitive data from one system to another. For example, in the operation of Close Air Support (CAS), commanders order voiced or written CAS requests to a ground guide. Then he has to control supporting bombers to the target area via voice, too. For the automated and networked operation process, we accelerate this process by relieving the need of human intervention. CoT aims to ensure the interoperable communication between Department of Defense (DoD) enterprises and increase the speed of operation.

The basic concept behind CoT is that its structure has to be as simple as possible while it has room to hold key information as well as surplus information for Communities of Interests (COI). Therefore, it has only three elements with 12 attributes for simplicity. On the other hand, CoT’s 12 required attributes represent time-sensitive WWW data for accuracy and generality. The main schema of CoT is shown in Table 2-1.

Elements	attributes	descriptions	WWW
event	version	Schema version: stable at 2.0	what
	type	Friendly tank? Hostile target?	what
	access (optional)		
	qos (optional)		
	opex (optional)		
	uid	Unique ID	what
	time	Time event was generated	when
	start	Valid start time	when

	stale	Valid interval time from start time	when
	how	Machine generated / human generated event ?	
point	lat	Latitude based on WGS84 in decimal degree	where
	lon	Longitude based on WGS84 in decimal degree	where
	hae	Height above ellipsoid based on WGS84 in meter	where
	ce	Circular error about point in meter	where
	le	Linear error about hae in meter	where
detail		Auxiliary space for COI	

Table 2-1 CoT 2.0 main schema

CoT sets up mandatory formats for 12 attributes but leaves detail deregulated. COI is able to implement detail in accordance with its intention.

[22] expands its structure to fit itself into the webservice in case of we have more than one target. The version 3.0 is now beta version but it is an appropriate topic because we try to handle the information exchange process in SOA concept. It puts elements as parent over the three elements: **details** above **detail**, **events** above **event**, and **points** above **point**. As a result, **details** can contain multi-detail, **events** contains multi-event, **points** holds multi-point. In addition, other attributes are introduced into the elements. The entire elements and attributes are shown in Table 2-2.

Elements	Attributes	Descriptions	WWW
events		Collection of event	
	version	Schema version: 3.0 beta	what
	access (optional)		
	opex (optional)		
event	id	Pointer used in one document	
	elementOrder	Ordering index	
	type	Friendly tank? Hostile target?	what
	qos (optional)		
	uid	Unique ID	what
	time	Time event was generated	when
	stale	Valid interval time from start time	when
	how	Machine generated / human generated event ?	
points		Collection of point	
point	id	Pointer used in one document	
	elementOrder	Ordering index	
	lat	Latitude based on WGS84 in decimal degree	where
	lon	Longitude based on WGS84 in decimal degree	where
	hae	Height above ellipsoid based on WGS84 in meter	where
	ce	Circular error about point in meter	where
	le	Linear error about hae in meter	where
details		Collection of detail	
detail		Auxiliary space for COI	
	id	Pointer used in one document	
	elementOrder	Ordering index	

Table 2-2 CoT 3.0 beta schema

*Id* is used as a reference pointer to indicate the element like point from another element like detail in one single CoT XML document. *ElementOrder* is an index of order of the

element in the common parent. Figure 2-11 shows all the relationships between **points**, **details**, and **events** in [27].

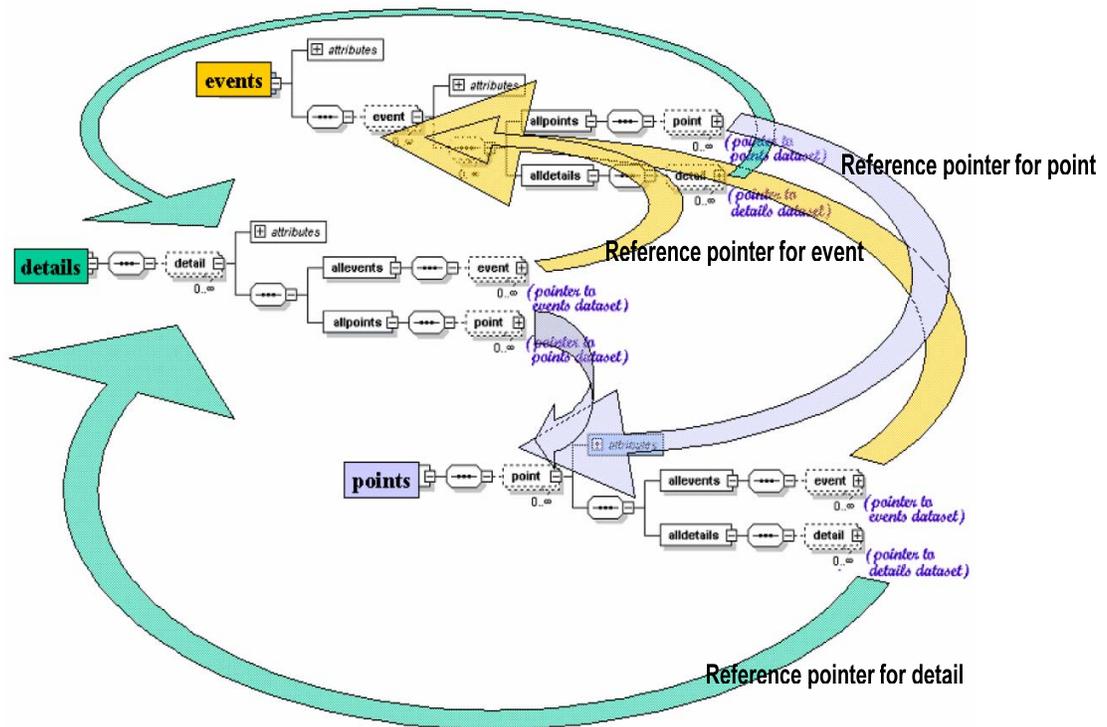


Figure 2-11 CoT 3.0 structure and relationships among elements. [27]

## 2.6. Battle Management Language (BML)

BML is defined as *an unambiguous language used to command and control forces and equipment conducting military operations and to provide for situational awareness and shared, common operational pictures* [16][18][19]. It has the following principles [16]:

- BML must be unambiguous;
- BML must not constrain the full expression of a commander's intent;
- BML must use the existing C4ISR data representations when possible;

- BML must allow all elements to communicate information pertaining to themselves, their mission and their environment in order to create an operational picture.

BML is a military communication language to bridge between real C2 systems and simulated forces, and perhaps, robotic forces in the future. Figure 2-12 shows a diagram to illustrate the role of BML among different military systems. BML is originally dedicated to express commanders' intent, request, and command in formal grammar and enhance the interoperability between real and simulated systems. It also expresses reports to commanders in a formal fashion.

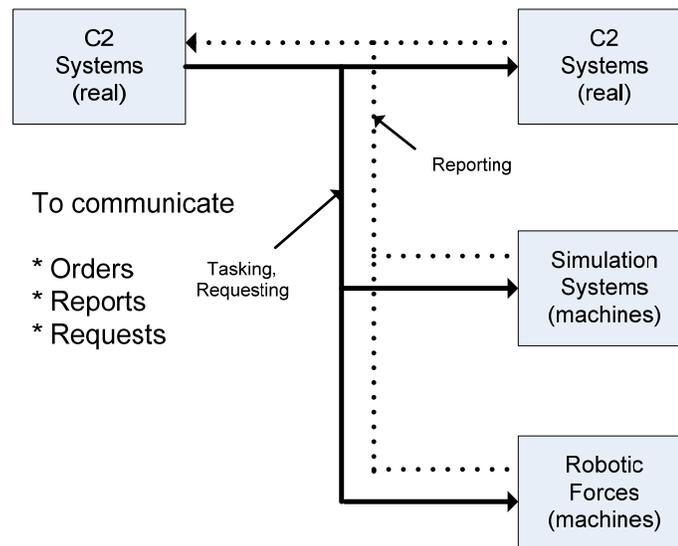


Figure 2-12 Diagram to illustrate the role of BML [16]

BML is developed based on several military standards, including Command and Control Information Exchange Data Model (C2IEDM) [50], and other US Army and US Marine Corps manuals including FM-101-5-1/MCRP 5-2A (Operational Terms and Graphics) [51] for doctrinal terms. It is now intended to extend to the international military operations called Coalition BML (C-BML) by adopting Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) [52] under Simulation Interoperability Standards Organization (SISO) as an effort to develop a standard [14].

[16] analyzes the BML, and characterizes it as a triangle with five sides: Protocols, Doctrine, Representation, Grammar, and Ontology. In this dissertation we consider all aspects of these BML characteristics in the Information Exchange Framework of SES ontology.

#### 2.6.1. BML grammar

BML grammar arises from the 5Ws (WHO, WHAT, WHERE, WHEN, and WHY) concept. The capability and limitation of the 5Ws as a BML grammar is evaluated in [18]. The advanced step to a formal grammar is presented in [18][20][21][22]. BML can be applied to various types of military communications. Orders, requests and reports are supported in the BML grammar.

Orders and requests have identical syntax, but the relation between taskers and taskees in the hierarchical rank of the military make them different. In an order, the tasker with

commanding authority mandates the task to the taskees, while the taskees are in the position of taking an order from the tasker in a request. The ordering and request syntax is shown as follows:

- $\text{OrderingParagraph} \rightarrow \text{CI OB}^* \text{C\_Sp}^* \text{C\_T}^*$  (Equation 2-6)

where **CI** is command intent, **OB** means ordering basic expressions for tasks, **C\_Sp** means spatial coordination expressions, and **C\_T** means temporal coordination expressions. The asterisk means there are arbitrarily many expressions for these parts.

The expansion rule for OB is given as:

- $\text{OB} \rightarrow \text{Verb Tasker Taskee (Affected | Action) Where}$   
 $\text{Start-When (End-When) Why Label (Mod)}^*$  (Equation 2-7)

**Verb** denotes task verbs, **Tasker** is names of the one who issues orders. **Taskee** means names of the units which take orders. **Affected**, someone is affected by the task, is determined whether it takes place by the choice of verbs. **Action** is also determined by the verb when another units' task is closely related to the verb. **Where** can be a location, or a path to a location. **Why** denotes a reason of the task. **Label** is a unique identification sign for the task. **Mod** is extra space for specific task.

Additionally, BML allows us to generate several types of reports such as task reports, event reports, status reports and position reports. Task reports are related with military activities. Event reports, on the other hand, include non-military activities beyond the task

reports. Status reports pertain to current situations of own, allied, and enemy troops. The basic syntax for reports is given by:

- ReportingParagraph  $\rightarrow$  RB\* (Equation 2-8)

where **RB** means reporting basic expressions.

The general expansion rules for **RB** is

- RB  $\rightarrow$  *task-report* Verb Executer (Affected | Action) Where When (Why) Certainty Label (Mod)\* (Equation 2-9)
- RB  $\rightarrow$  *event-report* EVerb (Affected | Action) Where When Certainly Label (Mod)\* (Equation 2-10)
- RB  $\rightarrow$  *status-report* Hostility Regarding (Identification Status-value) Where When Certainty Label (Mod)\* (Equation 2-11)

**Task reports** are reports on orders and **Verb** indicates the task verbs. **Executer** can be one of the followings: **Taskee**, **Agent**, or **Theme**. If a reporter knows the name of taskee it uses Taskee. If it does not know the name but only types, it uses **Agent**. **Theme** is used if only a reporter knows the main equipment of the executing unit. **Certainty** falls into one of the following: fact, plausible, uncertain, and indeterminate.

**Event reports** are reports about non-military events. **Everb** represents a specific event like a flood. **Status reports** include reports of position, general operational status, and status of equipment or personnel together with **Regarding**. In other words, **Regarding** denotes the topic of reports. **Identification** is relevant to the types of units or rank of

person. **Status-value** is related to operational status. Physical and duty status of a person are attached in this place as well.

This dissertation concerns the requests and status reports because the information exchange processes is not for tasking. The information comes from a higher or equivalent hierarchy as well as subordinate units. It is a process of ask and answer, not order and report. It is also about a shared common snapshot of a concerned area to understand current situation.

## CHAPTER 3. DATA FUSION PROCESS IN SES INFORMATION EXCHANGE FRAMEWORK

### 3.1. SES Ontology Implementation for Data Fusion Process

This section addresses some steps to implement SES ontology as an example of radar data representation. First, we define a master SES used in this dissertation, which shows a series of steps of implementation to carry out the pruning process from the computational aspect. We use SES builder for SES design environment. The SES Builder is a tool to design SES ontology from natural language style input [53].

The following language description rules are general forms of SES specifications [9]:

- **Specialization:**

THING can be VARIANT1, VARIANT2, or VARIANT3 in CLASSFAMILY !

- **Aspect:**

From VIEW perspective, THING is made of COMPONENT1, COMPONENT2, and COMPONENT3 !

- **Multi-aspect:**

From multiple perspective, THINGS are made of more than one THING !

- **Attached Variables:**

THING has VAR1, VAR2, and VAR3 !

- **Range Specifications of Variables:**

The range of THING's VAR1 is RANGE !

The capital letters represent variables which a user can specify. The others are reserved words and mandatory expressions. For example, the description in natural language style for Radar-SES is shown in Figure 3-1 and the whole natural language style description is attached in Appendix 1.

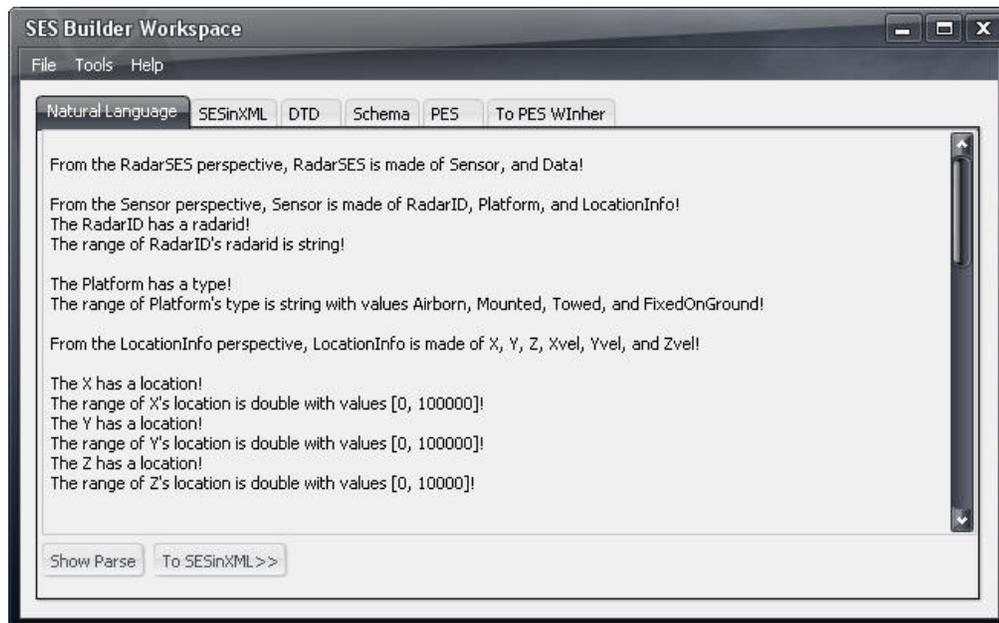


Figure 3-1 Natural language style description in SES Builder

More details on syntax and semantics are found in [9][12].

Subsequently, the designed SES in Figure 3-1 is displayed in a tree structure diagram such as Figure 3-2.

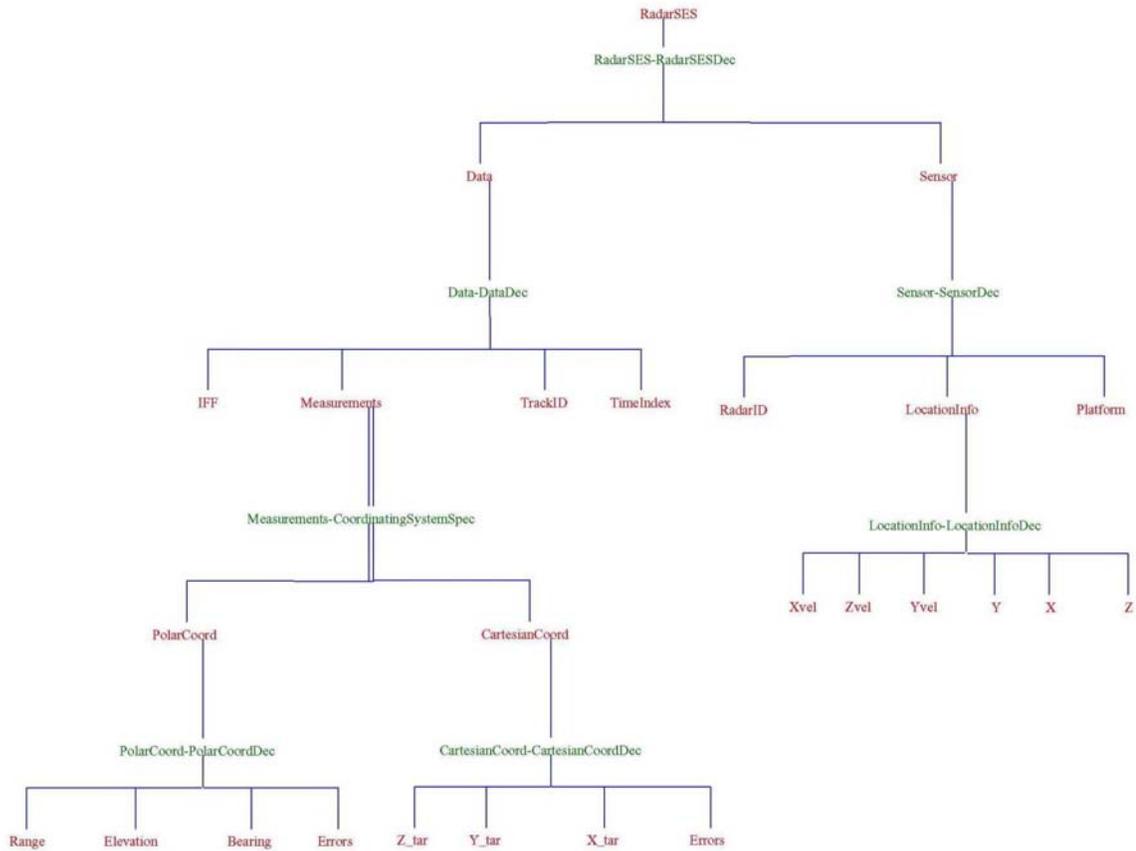


Figure 3-2 SES tree representation of radar data

Another output of SES design from a computational perspective is XML schema instances in DTD or Schema format [54][55]. These schema instances contain structural information for XML documents, which are outputs of SES ontological operations. The Schema of Figure 3-2 is shown in Appendix II.

An XML Schema holds structural information for XML documents [54]. The SES ontology is represented in XML format and instantiated as XML documents [9]. Therefore, an XML Schema reflects a SES structure.

In the master RadarSES in Figure 3-2, we can find two choices of measurements by coordinate systems. That is specialization relation between **Measurements** entity and sub-entities. The measurement type is determined by radar characteristics. If we choose the Cartesian coordinate system for simplicity, Figure 3-3 shows the result of pruning.

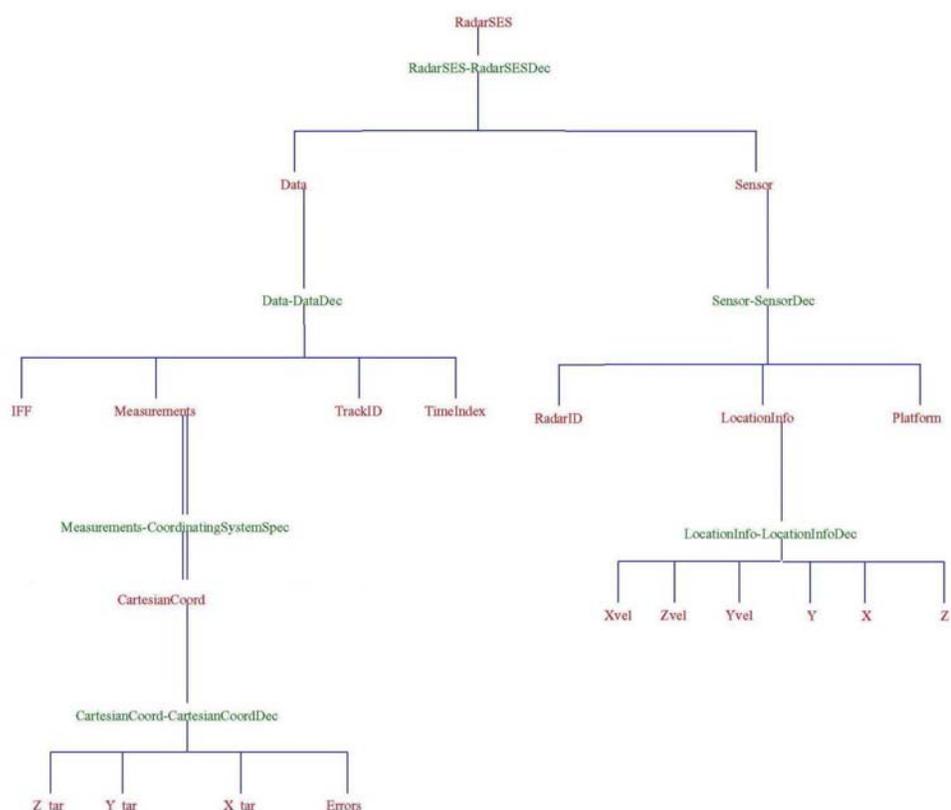


Figure 3-3 A pruned Radar-SES

This pruned SES is a data structure used in sensors to store their observation data in XML document files.

If we import this Schema of pruned sub-SES into the DEVJAVA modeling environment [56], we can manipulate the SES structure in schema format and generate XML files after data-binding with sensor data. A XML document after pruning and data-binding in sensors is shown in Appendix 3. Now we can store and extract some data from these XML files which are exchanged through the networks.

### 3.2. BML for Pragmatic Frame

The BML is extended to embody the pragmatic frames of IEF for the Data Fusion process. The commanders use BML to express their intent and orders [18][20], which are executed according to intended semantics by machines. That shows that BML can be understood by both human and machine. As we indicated before, the SES ontology includes pruning by pragmatic frames which specify information like a consumer's intent or request. In the military domain, consumers' requests can be represented by BML so that BML drives the message exchange processes between C2 systems and simulated systems.

Figure 3-4 illustrates how BML works as a pragmatic frame. BML describes a commander's intent including requests; it invokes pruning of master BML-SES. It produces three types of pruned BML-SES according to a commander's request of information level. The sub-SES is encoded in XML Schema format. After a DF system recognizes the commander's request from the XML Schema, it calls up a DF process of the request information. The result of DF process comes back as a report within the XML document, whose structure is defined in the sub-SES XML Schema.

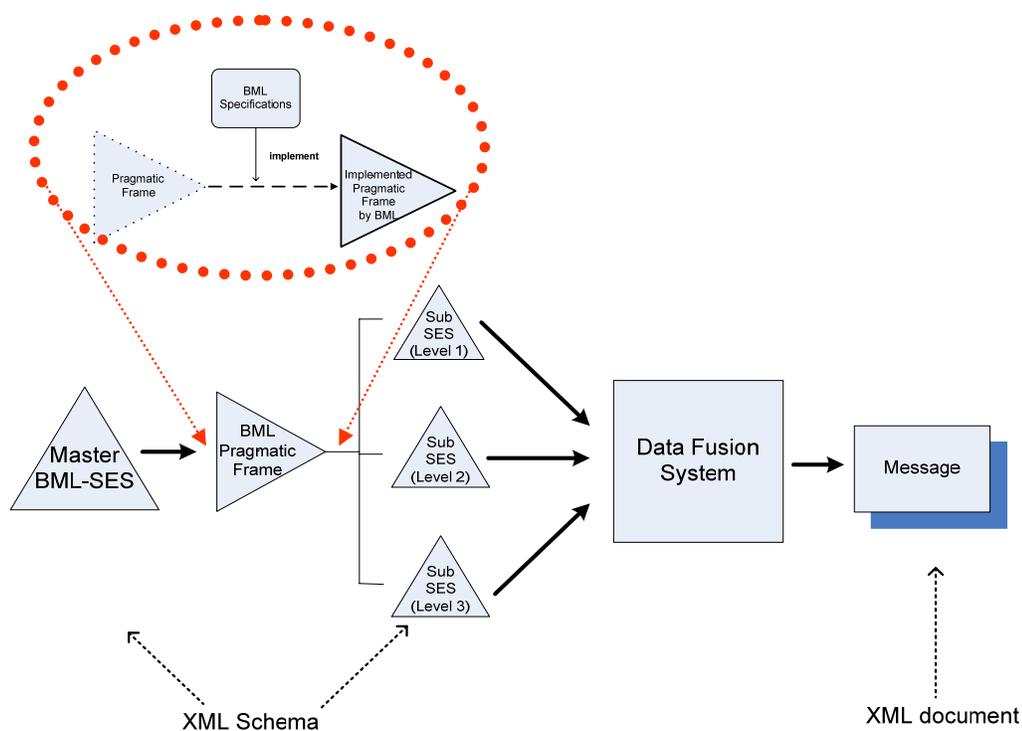


Figure 3-4 BML as a pragmatic frame

### 3.2.1. Basic BML implementation in SES ontology

BML is a formalized language. It has a formal grammar and standard lexicon called Command and Control Lexical Grammar (C2LG) so that it is easy to represent its structure in SES from the paragraph [17].

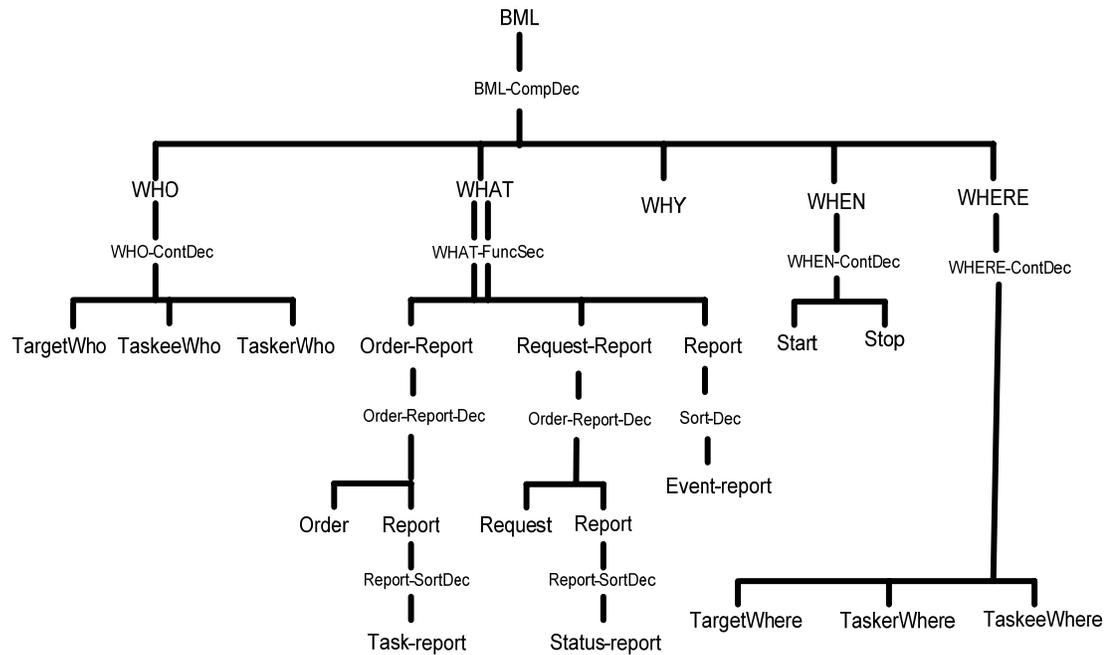


Figure 3-5 Basic BML representation in SES (not revised version)

Figure 3-5 shows a representation of BML in SES concept without extension for this dissertation. This master SES is changed by a commanders' order. When a commander places an order exemplified in [18], the pruned SES is shown in Figure 3-6.

- **march** MND-West(SP) M\_BDE13(NL) along DUCK start at Phase1A label\_3\_11

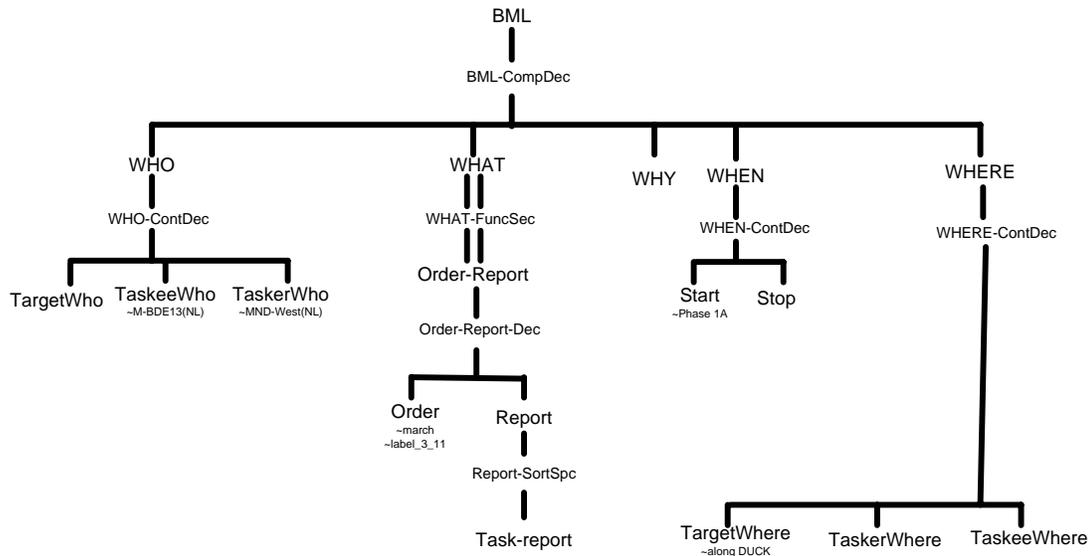


Figure 3-6 A pruned BML with an order

Tasker and taskee ids are attached to **TaskerWho** and **TaskeeWho**. To recognize if this pruned BML contains an order, the taskverb and the label are attached under **Order**. **Start** time and **Where** are attached as Figure 3-6.

### 3.2.2. Revision of BML paragraph for pragmatic frame

The current BML grammar for request is not fully suitable for our intention. **Verb** part of OB takes a role of WHAT in the 5Ws. However, it needs to be more specific so as to describe what taskees have to report back. The following is a revised version of a request from ordering reports form in [18].

- OB → *report* Tasker Taskee Hostility Regarding (At-Where) Start-When  
(End-When) Why Label (Equation 3-1)

This is a basic rule to give an order of report to a taskee. It is a variation of Equation 2-5. However, we want to modify it for information request. Consequently, we revise Equation 3-1 along with Equation 2-11.

- OB → *request* Tasker Taskee (Affected | Action) Regarding  
Interest-Where (Tasker-At-Where) Start-When (End-When) (Interval-When)  
Why Label (Mod)\* (Equation 3-2)

**Request** is a reserved word for a type of request instead of **report**. There is a difference between request and report. The former is used in a relation of which one sends a message to another which is not its subordinate. Conversely, the latter is used in a hierarchy. **Regarding** contains the contents of a report. For example, **Regarding** can be one of the following for each level of information of air objects.

- AirTargetsInfo: Level 1 info.
- AirSituation: Level 2 info.
- AirThreat: Level 3 info.

Since **Where** describes only **Interest-Where** in original BML grammar paragraphs, we insert **Tasker-At-Where** for Level 2 and Level 3 requests. In addition, **Interest-Where** should not be **Interest-At-Where** in some cases. This request for information is interested in multiple objects in a certain area. Another additional part is **Interval-When**, which tells update time for the next information. The basic update time follows the DF system's update processing time. If we don't write this part, the DF system will not return

the next information. Updated information could share some amount of information with the previous one in many cases. Therefore, we can update only new or changed parts of the information to relieve the communication load in a real system implementation. We suggest an extended BML-SES, which contains all the components of BML paragraphs in Figure 3-7, and its Schema is presented in Appendix IV.

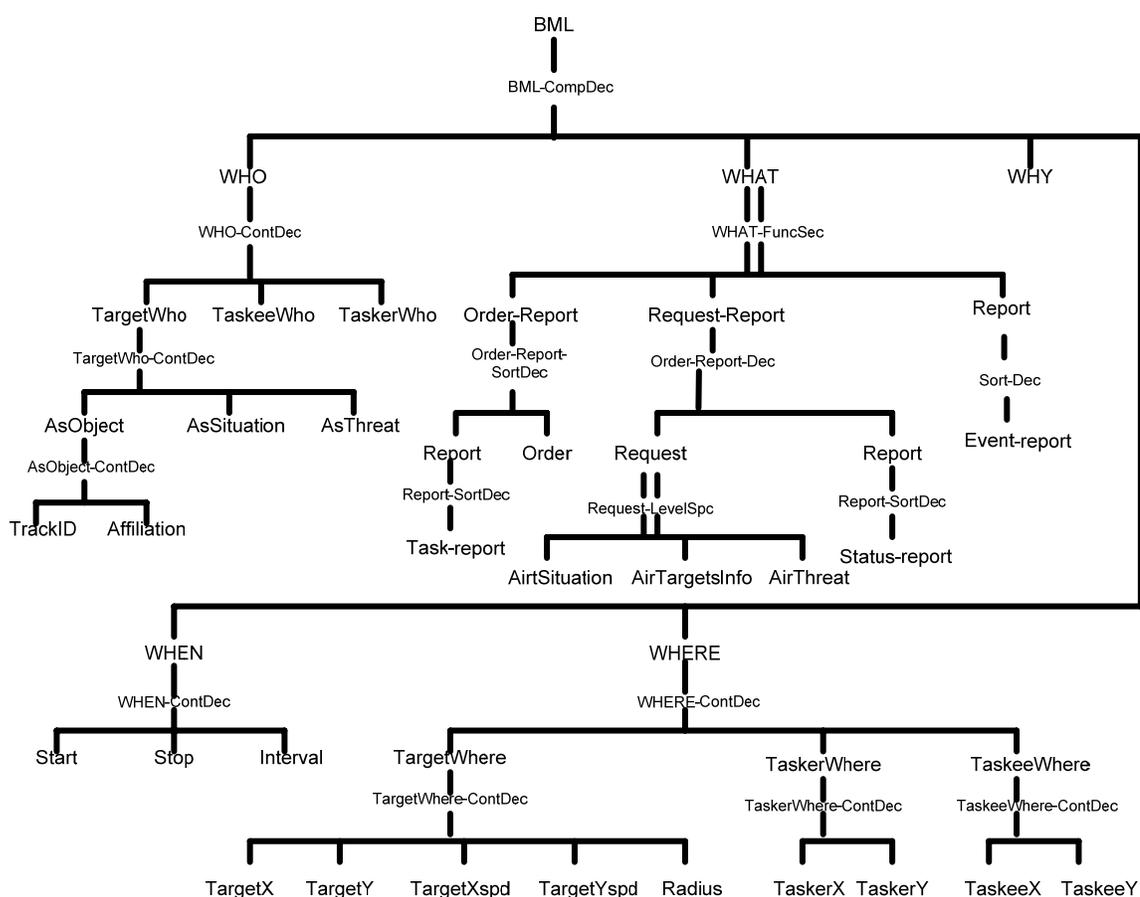


Figure 3-7 Extended SES for BML

A modification of report grammar is also necessary because we need to accommodate high level information in the paragraph.

- RB→ *status-report* Hostility (Relations/Situation) (Threat) Regarding (Identification Status-value) Target-At-Where When Certainty Label (Mod)\*  
(Equation 3-3)

The **Regarding** can be what level of information the report contains as mentioned in request as well as what it is intended in the original status report.

- AirTargetsInfo: Level 1 info.
- AirSituation: Level 2 info.
- AirThreat: Level 3 info.

**Relations** and **Threat** are used for containing results of Level 2 and Level 3 information.

The target location information is included in **Target-At-Where**.

### 3.3. High-level Information and Pragmatic Frame

The information that commanders request includes not only simple object information (Level 1) but also higher level derivations (Level 2 and Level 3). The more refined information is closely related to the relationships between users and targets. Such relations are defined by features such as: relative distance, target velocity, and targets moving direction. For the higher level information, users have to give their own information as well as specific requirements: user locations and the level of information that they expect from the information service providers. More generally, the user roles in the DF process have been deeply considered; Level 5, called User Refinement, is suggested in [31][57][58][59]. They emphasize human intervention in the DF process. Several functions of User Refinement are: Planning, Organizing, Controlling, Directing,

and Coordinating. More details are addressed in [58]. [31] suggests prioritization of need by human refinement in an ontological way. It is a similar way to the pragmatic frame, since it reduces the set of data which users require.

### 3.3.1. More SESs for SA of high-level information

For the automated reasoning process of SA in IEF, we need to define more SES ontology for high-level information description. We define Relation-SES and Threat-SES for Level 2 and Level 3. [60] shows a way to drive ontological meaning from the kinematics of targets. It focuses on kinematic relations between targets. [61] takes fuzzy-based approach for air defense operations. We modify and extend the ontology for relations between targets and users. Figure 3-8 shows a UserTargetRelation-SES for SA.

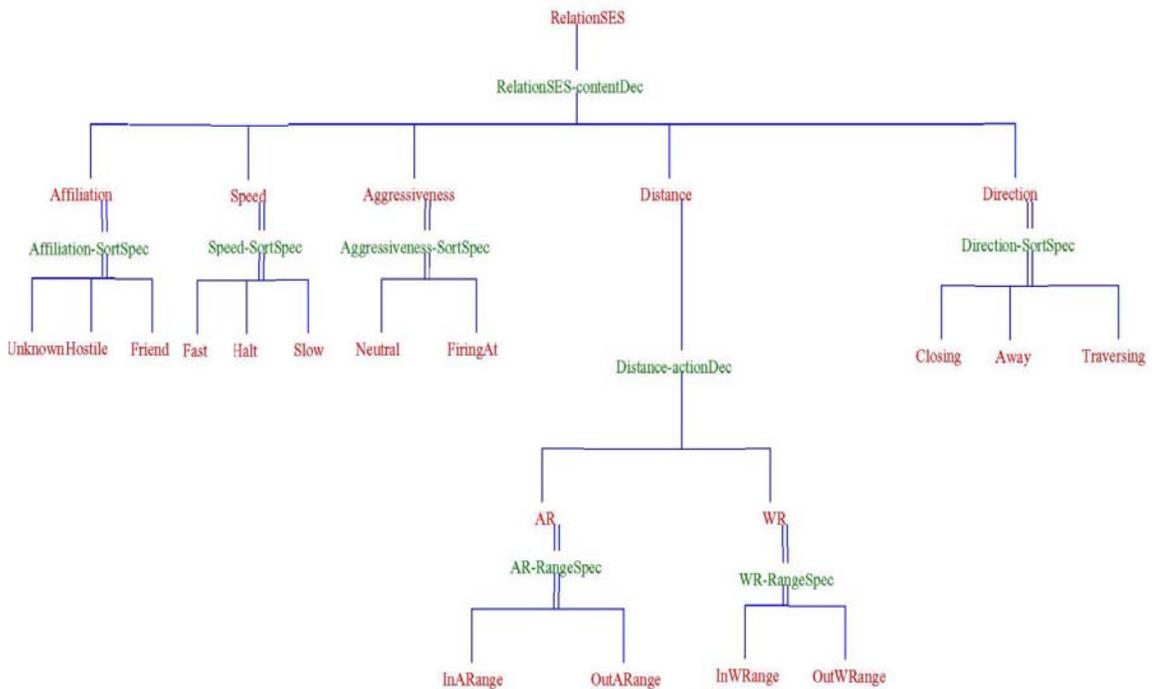


Figure 3-8 UserTargetRelation-SES

We have six Relations in UserTargetRelation-SES. Relations are drawn from features of targets:

- Speed: velocity of targets

Since we consider 2D space, we pick up two velocity vectors in X-axis and Y-axis.

Overall speed is calculated by:

$$velocity = \sqrt{\dot{x}^2 + \dot{y}^2} \quad (\text{m/s}) \quad (\text{Equation 3-4})$$

We assume slow target's velocity is less than 150m/s and more than 10 m/s, fast target's velocity is more than 150 m/s, and a halt target is assumed if it has less than 10 m/s of velocity.

- Direction: Relative direction determined by positions of targets and users.

We have to think about two angles for relative direction. The first one is target heading angle, which is the direction of target movement. The other is the angle of Line of Sight (LoS), which is an angle between a user and a target.

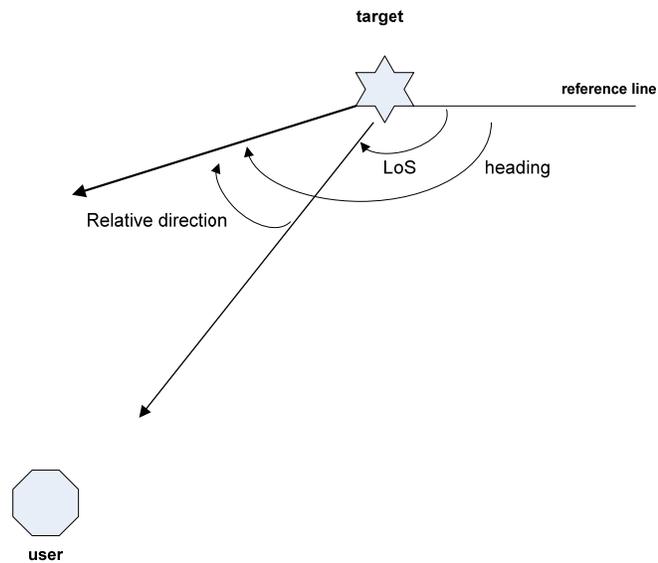


Figure 3-9 Illustration of relative direction

$$LOS = \arctan\left(\frac{y_{target} - y_{user}}{x_{target} - x_{user}}\right), \quad (\text{Equation 3-5})$$

$$heading = \arctan\left(\frac{\Delta y_{target}}{\Delta x_{target}}\right) = \arctan\left(\frac{\dot{y}_{target}}{\dot{x}_{target}}\right), \quad (\text{Equation 3-6})$$

$$direction\ angle = heading - LOS \quad (\text{Equation 3-7})$$

The relative direction is counted by subtracting.

We assume the three direction descriptions, and it can be obtained after adjusting the range of direction within  $[-180\ 180]$ . Figure 3-10 depicts relative direction descriptions:

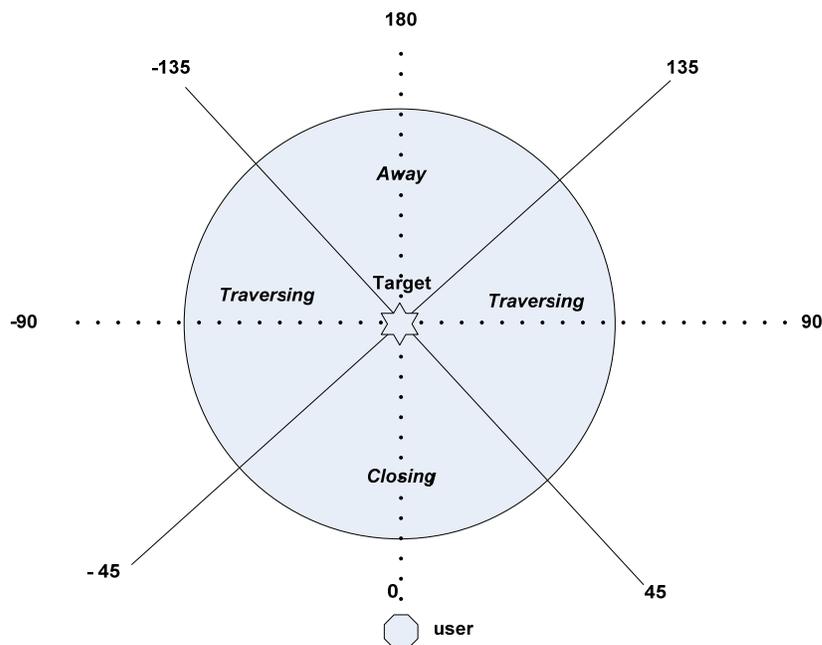


Figure 3-10 Relationships between relative direction and Direction entities in UserTargetRelation-SES.

If the absolute value of the direction angle is less than 45 degrees, then it is a closing target. If the absolute value of direction is less than 135 degrees and more than 45 degrees, it is traversing. If the absolute value of direction is more than 135 degrees to 180 degrees, it is getting away.

- Distance: Relative distance length between a target and a user.

We assume there are two reference lines for operations: Warning Range and Action Range.

The boundaries are determined by the objective of lines. The Warning Range boundary is established to give an early warning to all the units for preparation of the target. The

Action Range boundary is set up to do proper action against the target, for example by ordering engage.

- Affiliation is obtained by the target's affiliation information, such as an IFF notice.
- Aggressiveness is updated by either the sensor's report or other reports from other sources.

Figure 3-11 shows Threat-SES for Level 3.

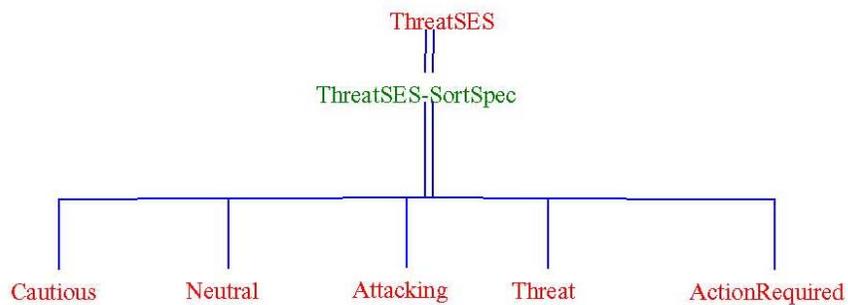


Figure 3-11 Threat-SES

For selection of Threat entities by reasoning, we collect all relations and compare the relations with pre-defined rules for threat types.

- “Action Required” can be driven by a collection of relations as follows:

If a target is { [fast (or) slow (or) halt] (and) [closing(or) traversing (or) away] (and) [Firing (or) Neutral ] (and) [Hostile] (and) [In ActionRange] (and) [In WarningRange] } then the target can be a {Action Required} target in the near future.

- “Attacking” target is described as:

If a target is { [fast (or) slow (or) halt] (and) [closing(or) traversing (or) away] (and) [Firing] (and) [Hostile] (and) [In ActionRange (or) Out ActionRange] (and) [In WarningRange (or) WarningRange] }, then the target can be a {Attacking} target now and in the near future.

- “Threat” can be driven by a collection of relations as follows:

If a target is { [fast (or) slow] (and) [closing(or) traversing] (and) [Firing (or) Neutral ] (and) [Hostile] (and) [Out ActionRange] (and) [In WarningRange] }, then the target can be a {Threat} target in the near future.

- “Cautious” can be driven by the following:

If a target is { [fast (or) slow] (and) [closing (or) traversing (or) away] (and) [Firing (or) Neutral ] (and) [Hostile (or) unknown] (and) [Out ActionRange] (and) [Out WarningRange] }, then the target can be a {Cautious} target in the near future.

- All other cases fall into “Neutral” target category.

### 3.3.2. A multi-level SES ontological DF process in IEF

The inference process for SA is converted to a mapping process in the SES ODF process in this dissertation. We have applied several mapping processes in the ODF process.

Mapping processes from BML paragraphs to the Schema of the master-SES are invoked when commanders or C2 systems place requests. We think of this mapping process as a pruning process in SES. A BML paragraph contains the choices for SES entities, which means it determines which entities have to be chosen or not. After the pruning or mapping step, a Schema instance of a sub-SES is generated, and it is sent to the local fusion center (service provider) with user information. As requests of the type come in to the center, it performs SES transformation from the BML-SES as in Figure 3-8 to the Radar-SES in Figure 3-2. Radar-SES describes the data of sensor systems. The transformation is another mapping process causing a pruning process in SES. The mapping relation of each entity of the two ontological representations is defined by using similar or same label names. A pruned Radar-SES is used as a reference to extract data from data base. The next step diverges by the requested information level. For the Level 1, inverse transformation occurs from Radar-SES to BML-SES and assigns the data to the entity variables of the sub-SES structure of BML-SES XML Schema. Then the XML Schema and combined data are converted to an XML document, which is returned to the requesting C2 system. On the other hand, for the more refined information, another pruning process of the Relation-SES is invoked by extracted features of data. The sub-SES of UserTargetRelation-SES, in turn, invokes a pruning process of Threat-SES in accordance with pre-defined rules.

The pruned relations, or threat, are attached under “AsSituation” or “AsThreat” entities of the pruned BML-SES. The Schema then is converted to an XML document and sent back

to the commander. In both cases, they become a report BML paragraph, which are displayed on the screen. The whole information exchange architecture is shown in Figure 3-12.

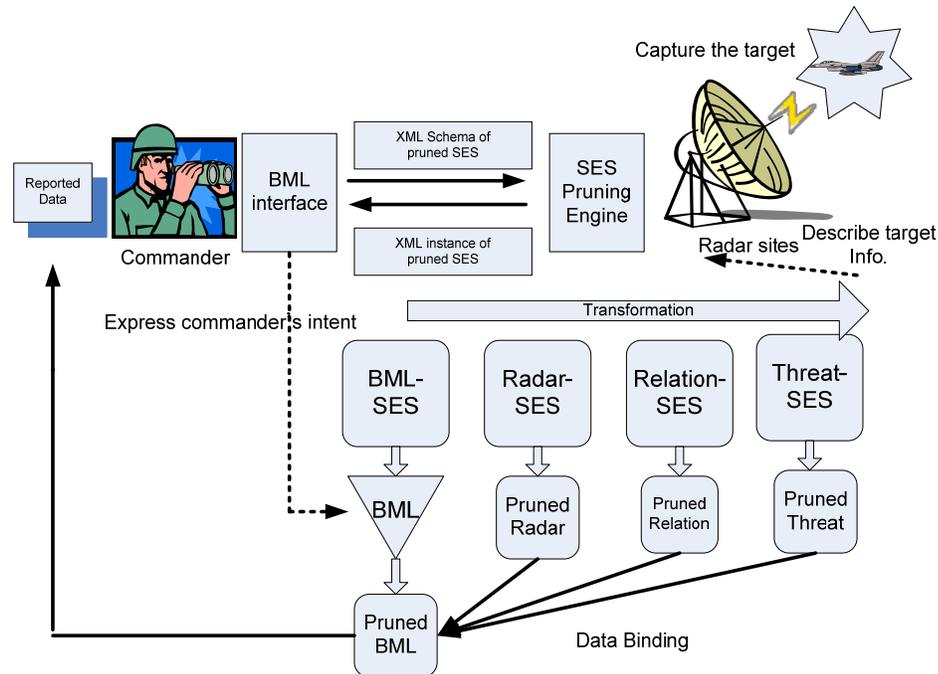


Figure 3-12. SES IEF via BML

### 3.4. Proof-of-Concept Examples

#### 3.4.1. Example 1

The commander of 01 battalion wants to receive continually updated basic information of air-targets concerning dangerous flying objects in the neighborhood of a point  $(X_p, Y_p)$  in the Cartesian coordinate system with radius of 4 miles to understand the current air space situation. He issues a request type to 001 radar site (fusion center) by formulating the BML request as follows:

- *request* 01Bat 001FC AirTargetsInfo  
at Xp, Yp with radius of 4 start at now continue label-r-001

“AirTargetsInfo” invokes the Level 1 process in sensor networks. If there is no interval indication, the report is going to send a report once. If it is said “continue” the information is updated in accordance with the sensor’s updating interval. The pruned BML-SES is shown in Figure 3-13. Xp and Yp represent the location of interest in the Cartesian coordinate system.

Accordingly, it produces a Schema that goes to the 001 radar site. The fusion center performs the specified transformation and pruning function based on the received Schema. The “TargetWho” and “TargetWhere” are related to the following entities of Radar-SES.

- TrackID / IFF/ X\_tar/ Y\_tar/ TimeIndex

At every update time, the Schema is bound with data and converted to an XML instance. This instance comes back to the commander as a text message of a BML report or tracks on the track display screen.

- *status-report* one hostile AirTargetsInfo  
at 30, 30 at now fact label-sr-001

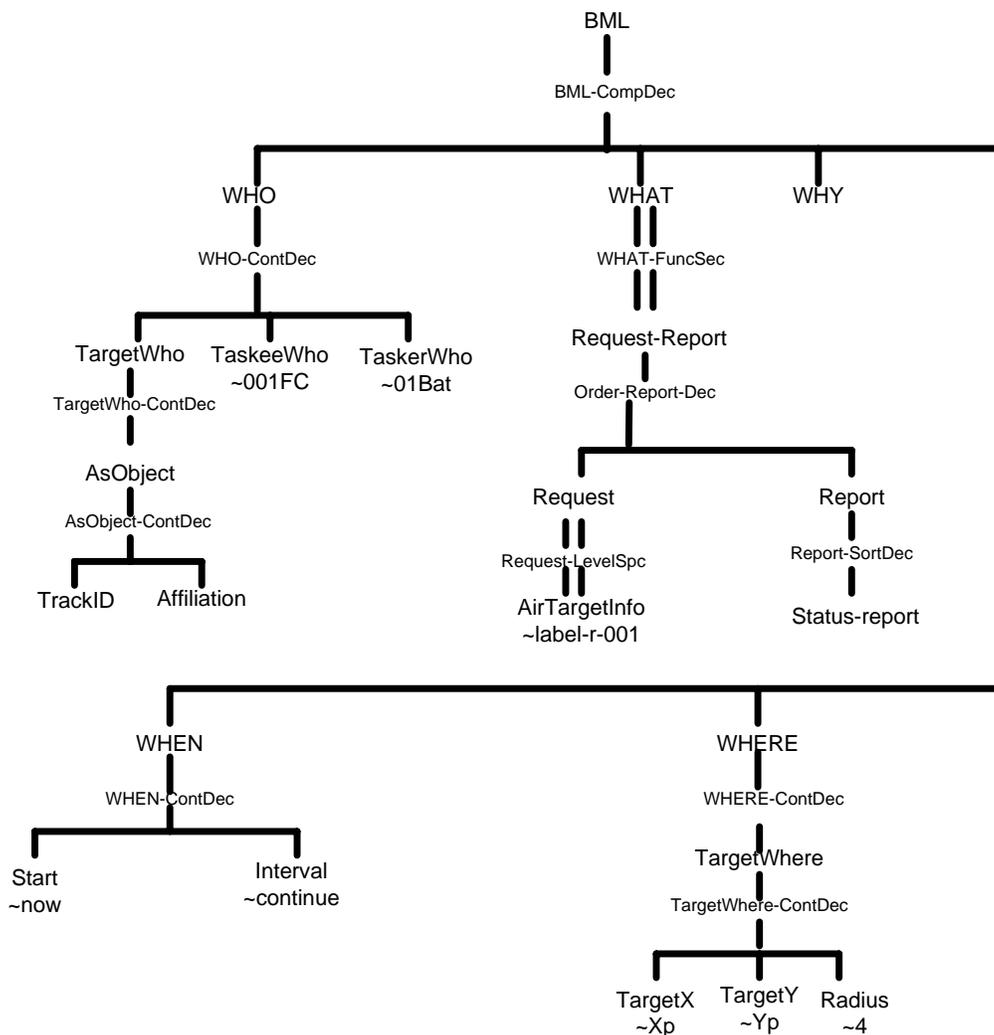


Figure 3-13. Pruned BML-SES for Level 1 information

### 3.4.2. Example 2

The same commander now wants to recognize threatening targets in the same area. He wants to determine whether or not he needs to turn the unit to yellow alert in accordance with the received threat analysis results. He puts a request using BML request as follows:

- *request* AirThreat 01Bat 001FC at Xp, Yp with radius of 4  
at Uxp, Uyp start at now continue label-r-002

“AirThreat” requests Level 3 information to sensor networks. Uxp, Uyp are, on the other hand, a user location in the Cartesian coordinate system. A pruned BML-SES is shown in Figure 3-14.

Based on relative distance between commander and targets, targets’ speed, targets’ heading, targets’ affiliation, and other event reports about those targets, we can infer the following relations from Relation-SES:

- A target is hostile, slow, neutral, away, out of warning range, out of action range from the commander.

A set of relations draws the following conclusion:

- The target is cautious.

It comes back to C2 system as a report:

- *status-report* one hostile cautious AirThreat  
at 32, 32 at now fact label-sr-002

Therefore, the commander could give an early notice of caution against the target; it will not be a warning.

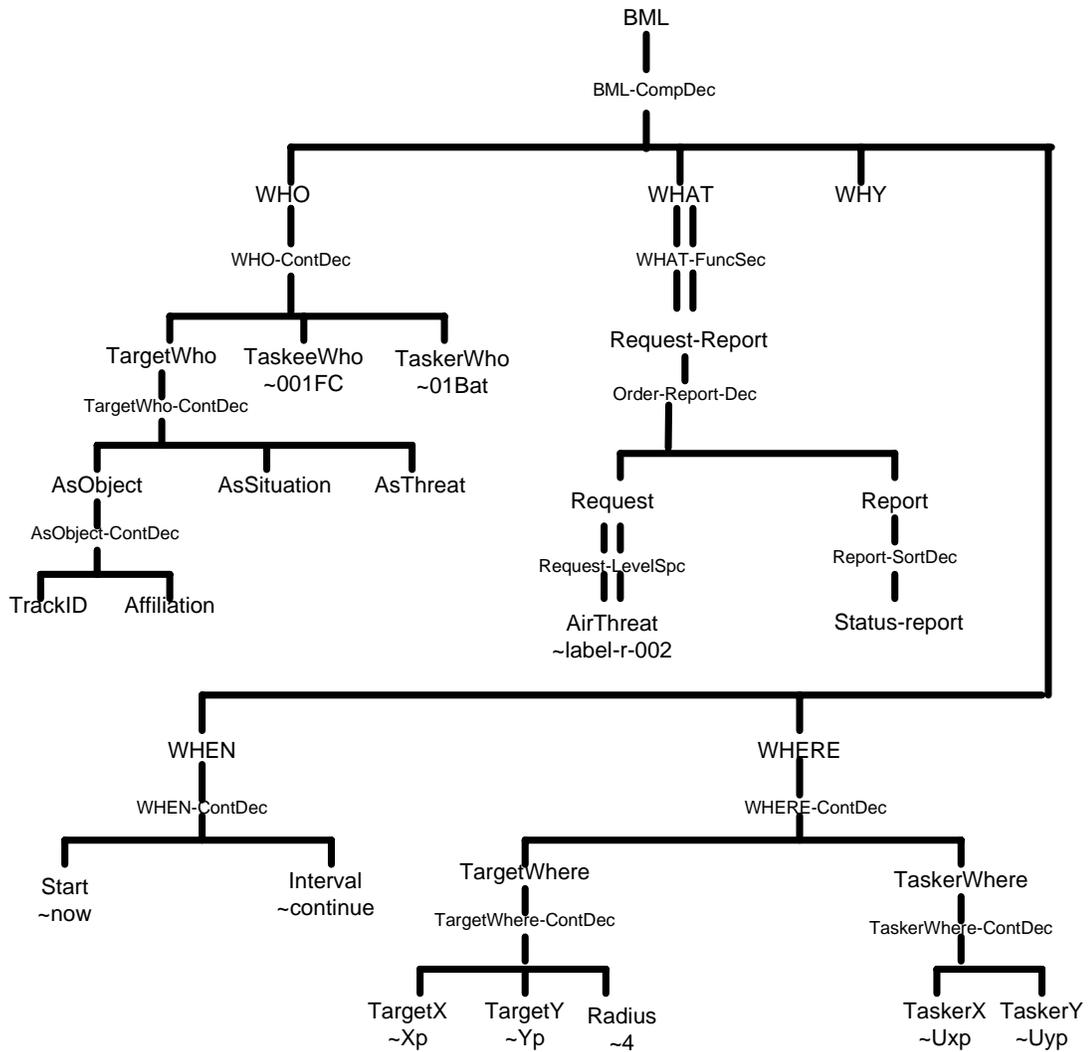


Figure 3-14 Pruned BML-SES for Level 3 information

## CHAPTER 4. AUXILLIARY ISSUES FOR LEVEL 1 DATA FUSION PROCESS

### 4.1. Multi-sensor Tracking with NN and Gating

There are two cases to use NN and gating in tracking problems in sensor networks.

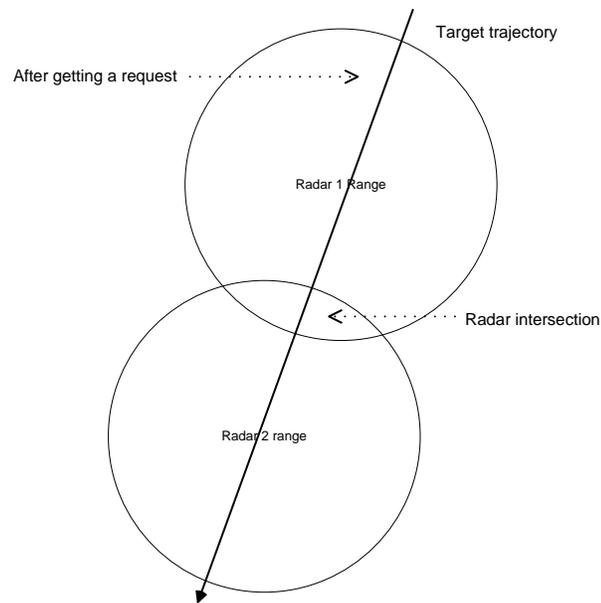


Figure 4.1 Two cases of ambiguous association.

The first case takes place when a sensor gets a request. It needs to find out appropriate targets with respect to the location of user interests. The second case is more complicated. When a target goes out of one sensor's coverage and comes into another sensor's boundary, the fusion center does not know which targets of the new sensors are related to the targets of former sensors which have been reported to users. The several methods to

find association between targets are used for this situation. An algorithm uses the correlation technique. However, it needs track history to calculate the correlation value. As a consequence, we need more time to store sufficient data after a target gets into the detection range, which causes delayed response to a request. Another way is to calculate “distance” and find the minimum one (Nearest Neighborhood) with gating [49]. We do not need to wait to collect data, since it just uses the most recent target location data. More sophisticated approaches are investigated in [47][48]. They are very powerful methods if high density of clutters or high probability of false alarms exists. However, we do not consider false alarm or clutters in the dissertation, since we have track data which are already filtered, and we assume no false targets. Hence, we choose the NN with the gating technique for the track association problem.

Gating is, in fact, a way to reduce candidates of association. We only need to check out the possibility of the relation among selected targets. Let us think about the situation that a fusion center has other track data, and it is required to find out the same target data from the collected track from sensors.

The first step is prediction to the current time from the last location of the target.

$$\hat{X}_{sensor1}(k/k-1) = F(k)\hat{X}_{sensor1}(k-1/k) \quad (\text{Equation 4-1})$$

$$\hat{P}_{sensor1}(k/k-1) = F(k-1)\hat{P}_{sensor1}(k-1/k-1)F(k-1)^T + Q(k-1) \quad (\text{Equation 4-2})$$

Sensor 1 is a sensor which has lost the target, since it exceeds the limits of the sensor detection area. The second one is to build up the volume of the gate centered at the predicted location:

$$[\hat{X}_{sensor1}(k/k-1) - \hat{X}_{sensor2}(k/k)]^T \hat{S}(k)^{-1} [\hat{X}_{sensor1}(k/k-1) - \hat{X}_{sensor2}(k/k)] < \kappa \quad (\text{Equation 4-3})$$

where

$$\begin{aligned} \hat{S}(k) = & H_{sensor1}(k) \hat{P}_{sensor1}(k/k-1) H_{sensor1}^T + R_{sensor1}(k) \\ & + H_{sensor2}(k) \hat{P}_{sensor2}(k/k) H_{sensor2}^T + R_{sensor2}(k) \end{aligned} \quad (\text{Equation 4-4})$$

The above Equation 4-3 and 4-4 come out by compensating for both track's uncertainty.

If a target of sensor 2, which is a new target, satisfies the Equation 4-3 and 4-4, we declare that the target is a candidate of the same target of sensor 1. If we have multiple targets in the volume of the gate, we choose the closest target.

We explain implementation of  $\hat{P}(\cdot)$ , which is an error term of estimates in Radar-SES.

$\hat{P}(\cdot)$  is formed in a matrix, and current Radar-SES does not have capability of representing a matrix. Hence, we choose MathML as a matrix representation format in XML document [62]. Mathematical Markup Language or MathML, in short, is an XML format for describing mathematical notation and capturing both its structure and content. The goal of MathML is to enable mathematics to be served, received, and processed over networks.

The basic format for Equation 4-5 is shown in the followings [62]:

- Matrix notation:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (\text{Equation 4-5})$$

- MathML description:

```

<mrow>
<apply>
  <eq/>
  <ci>A</ci>
  <matrix>
    <matrixrow>
      <ci>a</ci>
      <ci>b</ci>
    </matrixrow>
    <matrixrow>
      <ci>c</ci>
      <ci>d</ci>
    </matrixrow>
  </matrix>
</apply>
</mrow>

```

Figure 4-2 MathML representation of a matrix

Now we apply this format for representation of  $\hat{P}(\cdot)$  whose dimension is  $4 \times 4$ . The resulting representation is shown in Appendix I under Error entity.

## 4.2. Improve Interoperability between Radars with Cursor-on-Target (CoT)

Cursor-on-Target (CoT) is a XML-based message exchange enabler. It is so simplified that it just has 12 mandatory attributes, but other parts can be designed according to interest of users. We adopt CoT as a container to carry radar data to another radar for multi-sensor tracking. The **point** element contains key position information. It comes from sub-RadarSES's target location info. This location information is enough for the tracking, since main interests lie on the fuse of the location information in multiple sensors. However, for higher-level information, we need more information. That means we need to contain other information in CoT structure. This information is included in **details**. [28] shows a way to hold other information concerned by COI.

CoT 2.0 and CoT 3.0 beta are implemented in SES ontology as shown in Figure 4-3 and Figure 4-4.

CoT 3.0 is a little more complex than CoT 2.0 to manage multi-target information. **Refevent, refpoint, refdetail** are just kinds of pointers to indicate **details, events, points**. Original CoT3.0 has the same names of **event, point, and detail** for those pointers. We change the names to avoid violating the strict hierarchy of SES axioms [9]. The Schema for CoT 2.0 and CoT 3.0 beta are presented in Appedix VI, and Appendix V.

We intend to pass feature information over to another radar in CoT. Therefore, the sub-Radar-SES is attached under **detail** in the same way it is used as a container in [27][28].

The generated CoT XML file with RadarSES is conveyed to a fusion center, and it is used to produce combined information. A sample CoT XML message is shown in Appendix VI.

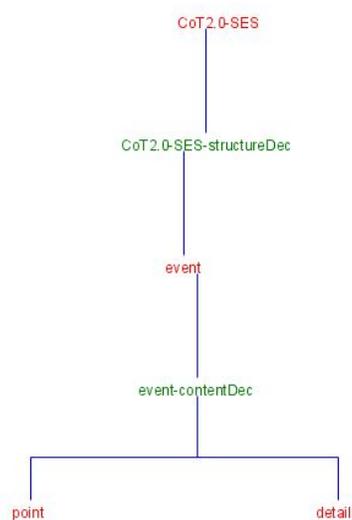


Figure 4-3 CoT 2.0-SES

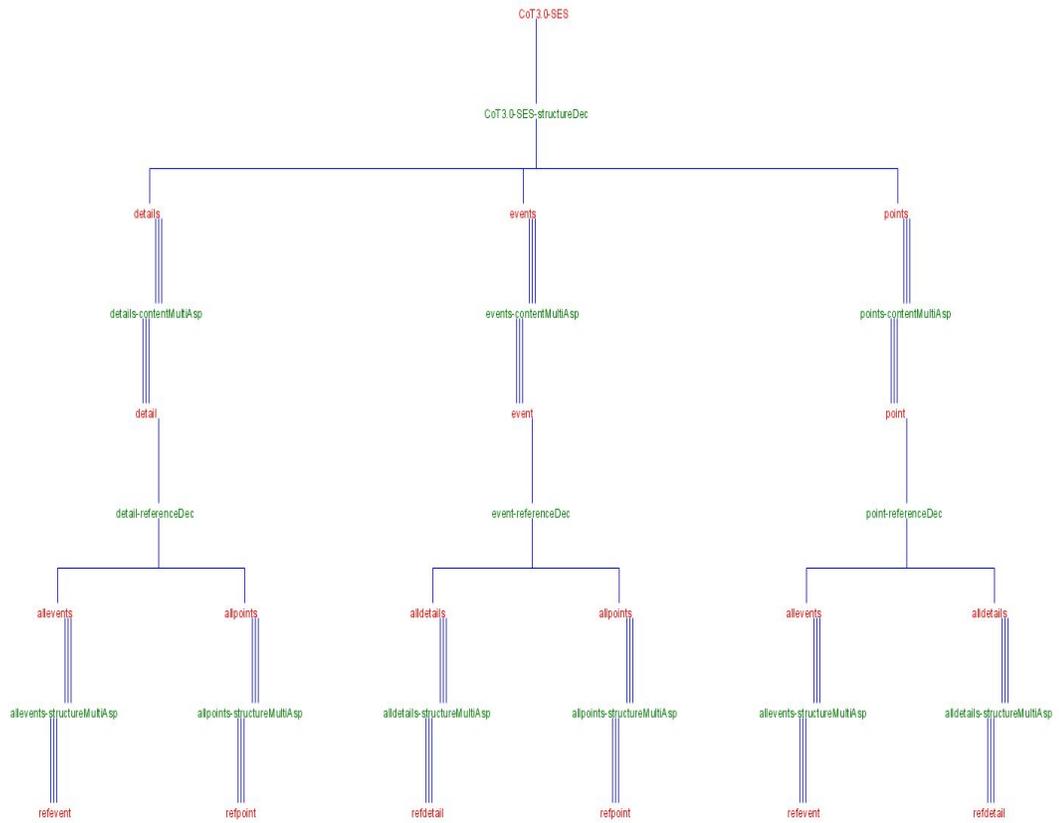


Figure 4-4 CoT 3.0 –SES

## CHAPTER 5. EVALUATION IN WARGAME SCENARIOS

The evaluation of the proposed SES ontology-based data fusion concept is carried out under three war-game scenarios. The first and second ones are intended to show a performance of the whole fusion process by showing multi-level information on the target. The third scenario applies additional specifications to show the effectiveness of this dissertation's approach by the result of a military operation.

### 5.1. Scenario 1-1

#### 5.1.1. Scenario description

An 101 Air-defense unit commander at Republic of Korea Army is ordered to be deployed to an area of responsibility for an air defense operation against North Korea's air-power. He requests air situation information to a local fusion center to plan the further operation. More detailed specifications for simulation are shown in Table 5-1:

Elements	Specification
commander	Located at (30km, 0)
Radar	Located at (30km, 30km)
radar detection range	A circle with radius of 60 km
Warning Range	70 Km from commander
Action Range	50 Km from commander
Target Trajectory	Start from (90km, 90km) Path: (a) – (b) – (c) – (d) – (e) –(f)

Table 5-1. Simulation specifications for scenario 1-1

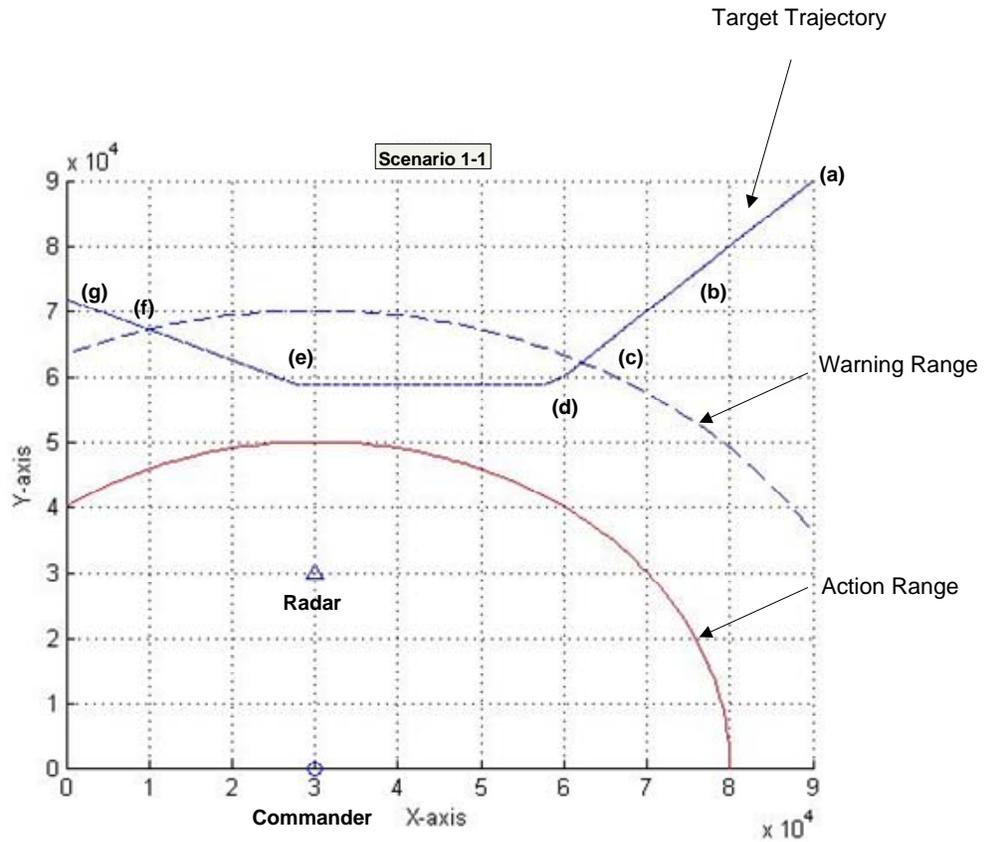


Figure 5-1 Scenario 1-1

### 5.1.2. DEVS models

Discrete Event System Specification (DEVS) is an advanced and well-defined mathematical modeling and simulation formalism based on system theory [63]. For decades, DEVS has been applied to diverse modeling and simulation problems with various extensions such as Dynamic Structure DEVS, Symbolic DEVS, Fuzzy DEVS, and Real-Time DEVS [63].

### 5.1.2.1. DEVS formalism

The formalism for an atomic model and a coupled model is shown below [63]:

- Atomic model:

$$M = \langle X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, \text{ta} \rangle \quad (\text{Equation 5-1})$$

where,

X: a set of inputs;

S: a set of states;

Y: a set of outputs;

$\delta_{\text{int}}$ : Internal transition function;

$\delta_{\text{ext}}$ : External transition function;

$\lambda$ : Output Function;

ta : Time advance function.

- Coupled model:

$$DN = \langle X, Y, D, \{M_i\}, \{I_i\}, \{Z_{i,j}\} \rangle \quad (\text{Equation 5-2})$$

where,

X: a set of external input events;

Y: a set of outputs;

D: a set of components names, for each i in D;

$M_i$ : a component model;

$I_i$ : the set of influences for I; for each j in  $I_i$ ;

$Z_{i,j}$ : the i-to-j output translation function.

We implement the DEVS models in the DEVSJAVA environment [56]. DEVSJAVA, which is a DEVS modeling and simulation environment in Java, supports the implementation of the various DEVS extended formalism through Simview. We use DEVSJAVA in Eclipse [64].

The first model for scenario is a scenario generator model. The model, in fact, generates real target location information with velocity and heading specifications. For example, a target's X –axis position and Y- axis position are calculated by Equation 5-3 and 5-4 in Cartesian coordinates.

$$x(k) = x(k-1) + \dot{x}(k-1) \times \cos(\theta) \times T \quad (\text{Equation 5-3})$$

$$y(k) = y(k-1) + \dot{y}(k-1) \times \sin(\theta) \times T \quad (\text{Equation 5-4})$$

where,  $\theta$  is heading in radian, which has positive value in counter-clock wise direction from reference line of East.  $T$  is time-interval of propagation in second.  $x(k)$  and  $y(k)$  represent locations on X-axis and Y-axis in Cartesian coordinate at time  $k$  in meter.  $\dot{x}(k-1)$  and  $\dot{y}(k-1)$  represent velocity on X-axis and Y-axis in Cartesian coordinate in meter/second.

The above equations are simple formulas to propagate location to the next time. Therefore, if we specify time, heading, and velocity, we can generate track data in a certain time length.

The specifications of target dynamics are described in XML document from DEVJSJAVA.

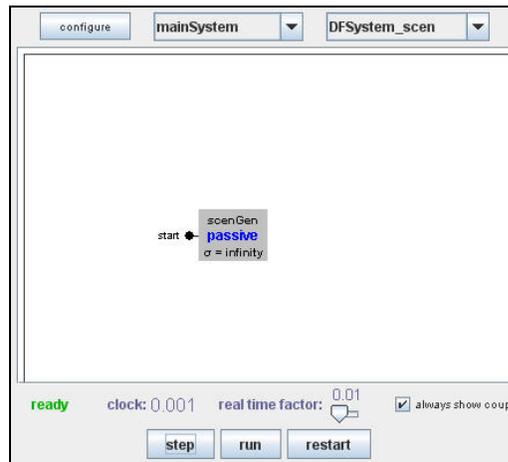


Figure 5-2 A scenario generator

```

Time: 0.0 ,input injected:
port: start value:
==== Scenario Generation =====:
time (in sec) ?
170
direction (in deg) ?
-135
velocity (in m/s) ?
250
Done ?
n
time (in sec) ?
180
direction (in deg) ?
-155
velocity (in m/s) ?
250
Done ?
n
time (in sec) ?
300
direction (in deg) ?
-180
velocity (in m/s) ?
250
Done ?
n
time (in sec) ?
500
direction (in deg) ?
155
velocity (in m/s) ?
250
Done ?
y
Terminated Normally before ITERATION 2 ,time: 0.0010

```

Figure 5-3 DEVJSJAVA input console for scenario specification

Figure 5-2 shows a DEVS model to generate a scenario specification. When we run the model, we can see the following messages on the console. If we enter some inputs like Figure 5-3, it turns them into a XML file in Figure 5-4:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<scenarios>
  <scenarios-multipleMultiAsp>
    <scenario heading="-135.0" time="170.0" velocity="250.0"/>
    <scenario heading="-155.25" time="180.0" velocity="250.0"/>
    <scenario heading="-180.0" time="300.0" velocity="250.0"/>
    <scenario heading="155.0" time="500.0" velocity="250.0"/>
  </scenarios-multipleMultiAsp>
</scenarios>
```

Figure 5-4 XML document output for scenario 1-1

The meaning of the specifications is:

- From 0 second to 170 second, a target moves toward -135 degree with 250m/s speed.
- From 171 second to 180 second, the target moves toward -155 degree with 250 m/s speed.
- From 181 second to 300 second, the target moves toward -180 degree with 250 m/s speed.
- From 301 second to 500 second, the target moves toward 155 degree with 250 m/s speed.

The XML file is a blue print to generate a target for scenario 1-1. Now we make several models to simulate scenario 1-1.

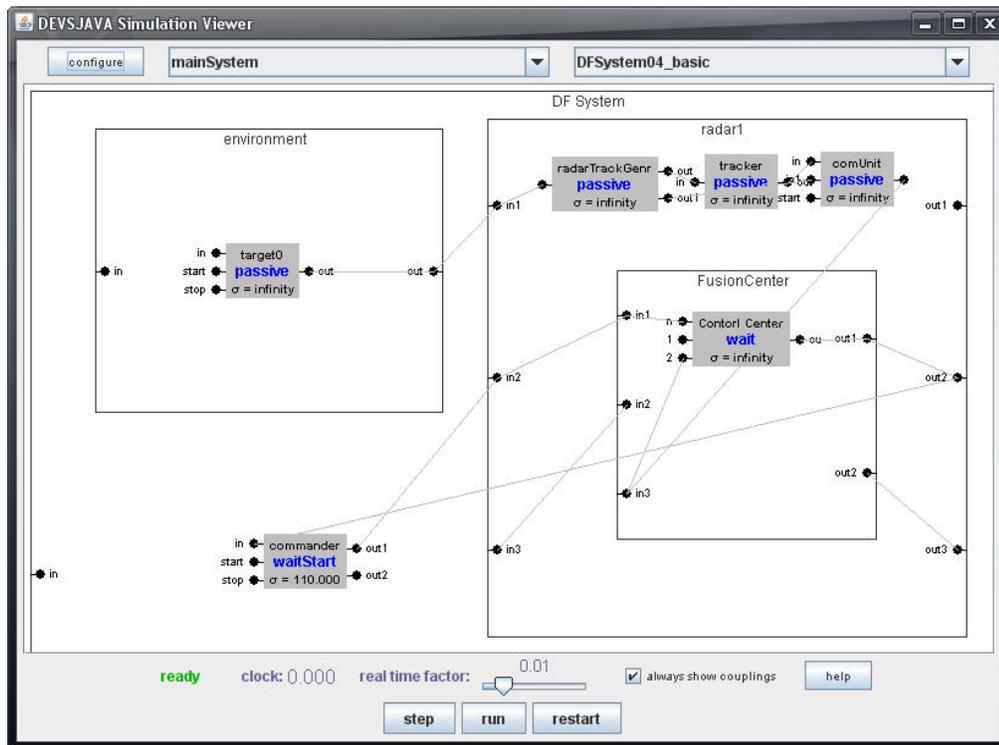


Figure 5-5 DEVSJAVA models in Simview.

The **target0** model imports a scenario XML file and generates target location information in accordance with Equation 5-3 and Equation 5-4.

The **radarTrackGenr** model adds measurement noise when the target is in detection range. The noise is derived from Gaussian distribution with 5 m of standard deviation.

The **tracker** model is a model implemented by the KF algorithm. To implement a KF, we define a state vector and other matrices described in Chapter 2.

$$X(k) = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix},$$

(Equation 5-5)

$$Y(k) = \begin{bmatrix} x \\ y \end{bmatrix},$$

(Equation 5-6)

$$F(k) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(Equation 5-7)

$$H(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

(Equation 5-8)

$$Q(k) = \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} \sigma_v^2,$$

(Equation 5-9)

$$R(k) = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix}$$

(Equation 5-10)

The state and measurement vectors are appropriate for representing Cartesian coordinate information, and state transition matrix and measurement matrix come from Kinetic models for linear modeling of target dynamics.  $\sigma_v^2$  is process noise which means error of modeling of target dynamics. We set it up at 200.  $\sigma_w^2$  means observation error. We set it up at 25.

KF needs a capability of matrix operation in DEVJSJAVA. We bring up the JAMA package for linear algebra operations in Java. JAMA, which was developed by the MathWorks and NIST, contains user-level classes to deal with real and dense matrices [65]. It is simple but supports full functionalities for implementation of KF.

The **tracker** model generates output in a recursive way, as shown in Figure 2-9. It generates XML files formatted by pruned Radar-SES. The file goes to the **comUnit** model and it converts the radar data into the CoT XML file.

The **ControlCenter** model is a control part in a fusion center to accept requests and respond to the requests with current radar data in CoT format.

The **commander** is an agent model to mimic the commander's behaviors for request and reports.

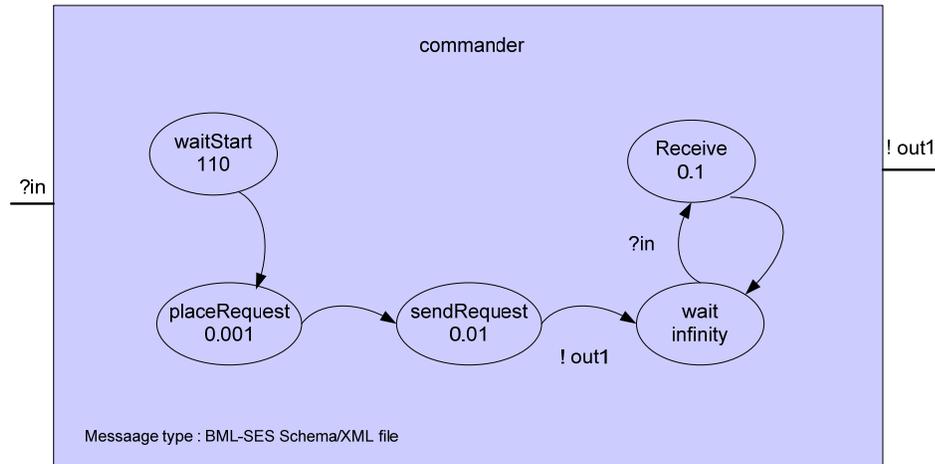


Figure 5-6 State diagram of **commander** model

The **commander** sends a request of Level 3 in BML format at 110 seconds after starting the simulation. It waits for report from sensors. When a report returns from sensor, the commander parses the status-report file and displays the target in the **TrackDisplay**.

The **TrackDisplay** is connected to the **commander** model. When a report comes with target data it shows them on the screen. We use OpenMap<sup>TM</sup> library to implement the display [66]. OpenMap<sup>TM</sup> is a free toolkit to implement applications with geospatial data provided by BBN technology. We can use this API without any restriction and add other functionalities in accordance with our intentions. A screenshot of the **TrackDisplay** is shown in Figure 5-7.

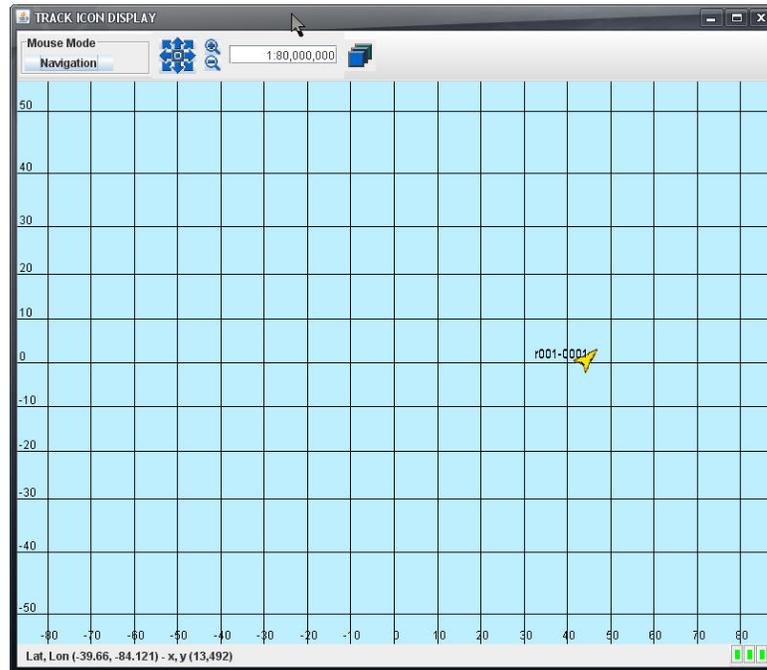


Figure 5-7 Snapshot of TrackDisplay

The arrow represents a target icon, and trackID is attached to it. For a look-and-feel of threat types, we assume several colors:

- Red : hostile
- Yellow: unknown, neutral
- Blue: Friend
- Black : Action required

OpenMap™ takes latitude, longitude, and altitude as arguments of location information. On the other hand, we also use Cartesian coordinates. We need to convert Cartesian coordinates into latitude, longitude, and altitude to display the target. The conversion algorithm is shown in Appendix VII.

## 5.1.3. The results of simulation under scenario 1-1

Target location	Relations	Threat type	Explanation
(a)	NaN	NaN	Target initial point
(b)	Fast,Closing,Neutral,Hostile, OutWRange,OutARange	Cautious	First response of request
(c)	Fast,Closing,Neutral,Hostile, InWRange,OutARange	Threat	Comes in Warning Range
(d)	Fast,Traversing,Neutral,Hostile, InWRange,OutARange	Threat	Change Direction
(e)	Fast,Traversing,Neutral,Hostile, InWRange,OutARange	Threat	Change Direction
(f)	Fast,Traversing,Neutral,Hostile, OutWRange,OutARange	Cautious	Go out of Warning Range
(g)	Fast,Away,Neutral,Hostile, OutWRange,OutARange	Neutral	Change Relative direction

Table 5-2 Changes of relations and threat types with respect to change of target states in scenario 1-1.

At (a) we do not receive any information because we did not request any yet. From (b) we receive information after giving a request. When it comes into the Warning Range, the threat type turns into threat. Even though the target changes its heading, it is still in Warning Range. Therefore, its threat type does not change until it gets out of Warning Range through (c) – (e). When the target flies away from the Warning Range, the threat

type becomes *Cautious* at (f). The relative direction shifts to another category when it gets away from user at (g).

## 5.2. Scenario 1-2

### 5.2.1. Scenario description

This scenario is basically the same as scenario 1-1 except using a different target trajectory. The output XML file is shown in Figure 5-8.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<scenarios>
  <scenarios-multipleMultiAsp>
    <scenario heading="-115.0" time="200.0" velocity="250.0"/>
    <scenario heading="-135.0" time="210.0" velocity="250.0"/>
    <scenario heading="-155.0" time="220.0" velocity="250.0"/>
    <scenario heading="180.0" time="300.0" velocity="250.0"/>
    <scenario heading="155.0" time="310.0" velocity="250.0"/>
    <scenario heading="135.0" time="320.0" velocity="250.0"/>
    <scenario heading="115.0" time="500.0" velocity="250.0"/>
  </scenarios-multipleMultiAsp>
</scenarios>
```

Figure 5-8 XML description of scenario 1-2

The scenario is illustrated in Table 5-3 and Figure 5-9:

elements	Specification
commander	Located at (30km, 0)
radar	Located at (30km, 30km)
radar detection range	A circle with radius of 60 km
Warning Range	70 Km from commander

Action Range	50 Km from commander
Target Trajectory	Start from (90km, 90km) Path: (a) – (b) – (c) – (d) – (e) –(f) – (g) – (h)

Table 5-3 Simulation specification of scenario 1-2.

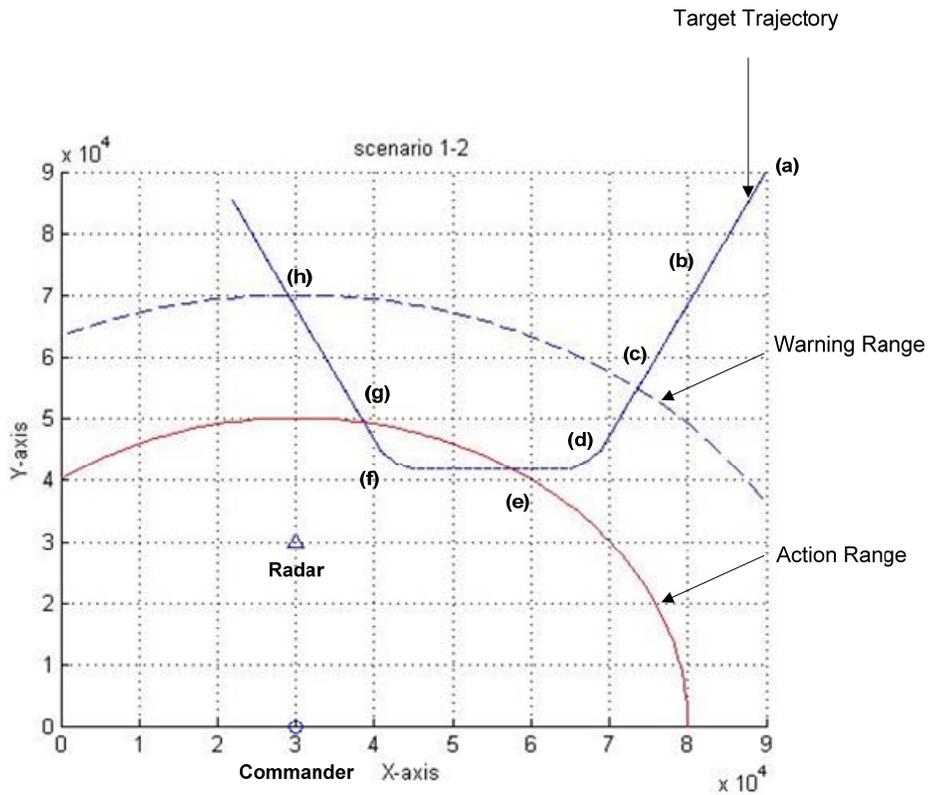


Figure 5-9 Illustration of scenario 1-2

The target comes closer than scenario 1-1 and gets across the Action Range boundary. Consequently, the threat type turns into *ActionRequired* as it comes over the line.

Now that we change the scenario, we alter the **commander** model's behavior as well to work under this scenario. Its first state, "waitRequest," remains until 130 seconds and places a request after that.

### 5.2.2. The result of simulation under scenario 1-2

The simulation result is shown in Table 5-4:

Target location	relations	Threat type	Explanation
(a)	<i>NaN</i>	<i>NaN</i>	Target initial point
(b)	Fast,Closing,Neutral,Hostile, OutWRange,OutARange	Cautious	First response of a request
(c)	Fast,Closing,Neutral,Hostile, InWRange,OutARange	Threat	Comes in Warning Range
(d)	Fast,Traversing,Neutral,Hostile, InWRange,OutARange	Threat	Change Direction
(e)	Fast,Traversing,Neutral,Hostile, InWRange,InARange	Action Required	Comes in Action Range
(f)	Fast,Away,Neutral,Hostile,InWR ange,InARange	Action Required	Change Direction
(g)	Fast,Away,Neutral,Hostile,InWR ange,OutARange	Cautious	Out of Action Range
(h)	Fast,Away,Neutral,Hostile,OutW Range,OutARange	Neutral	Out of Warning Range

Table 5-4 Changes of relations and treat types with respect to target states in scenario 1-2.

The target's information is sent to the user for the first time at (b). It gets to threat when it gets across the Warning Range at (c). The target keeps its threat type when it is in

Warning Range and moves across the area at (d) although it alters its heading. As it comes into Action Range, we must act against the target as it progresses from (e) to (f). As it goes out of the Action Range and gets away from the user, the target's threat type turns into *Cautious* at (g). Since it still moves away and is out of Warning Range at (h), the target becomes *Neutral* target.

### 5.3. Scenario 2

Scenario 2 contains more complex situations. The commander issues orders and requests. The commander can engage threat targets with an air defense unit. Two short range radar units are operating in the battlefield so that we can evaluate the multi-sensor tracking algorithm with NN and gating techniques as well.

#### 5.3.1. Scenario description

The 101 army regiment commander received an order to defend an area of responsibility. He has just received a warning from the higher echelon. It is highly probable that the enemy's air strikes to help their attack are imminent to the area. Against this threat, he warns all units to prepare for the air assaults. Fortunately, he has an air defense company under his command. He will command the unit to engage against the air threats as he decides.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<scenarios>
  <scenarios-multipleMultiAsp>
    <scenario heading="-135.0" time="600.0" velocity="250.0"/>
  </scenarios-multipleMultiAsp>
</scenarios>

```

Figure 5-10 XML description of scenario 2

Figure 5-10 shows that the target trajectory of scenario 2 is straightforward compared to previous ones. It moves to the diagonal direction toward the commander to easily determine the threat types of the target as shown in Figure 5-11.

The scenario specifications are listed in Table 5-5.

elements	Specification
commander	Located at (0, 0)
Air defense unit	Initially located at (0, 10km): point (A). After receiving an order of march, it moves to (10km,10km): point (B)
Radar1	Located at (20km, 20km)
Radar 2	Located at (60km, 60km)
radar detection range	A circle with radius of 30 km
Warning Range	70 Km from commander
Action Range	50 Km from commander
Target Trajectory	Start from (90km, 90km) Path: (a) – (b) – (c) – (d) – (e)

Table 5-5 Simulation specification of scenario 2

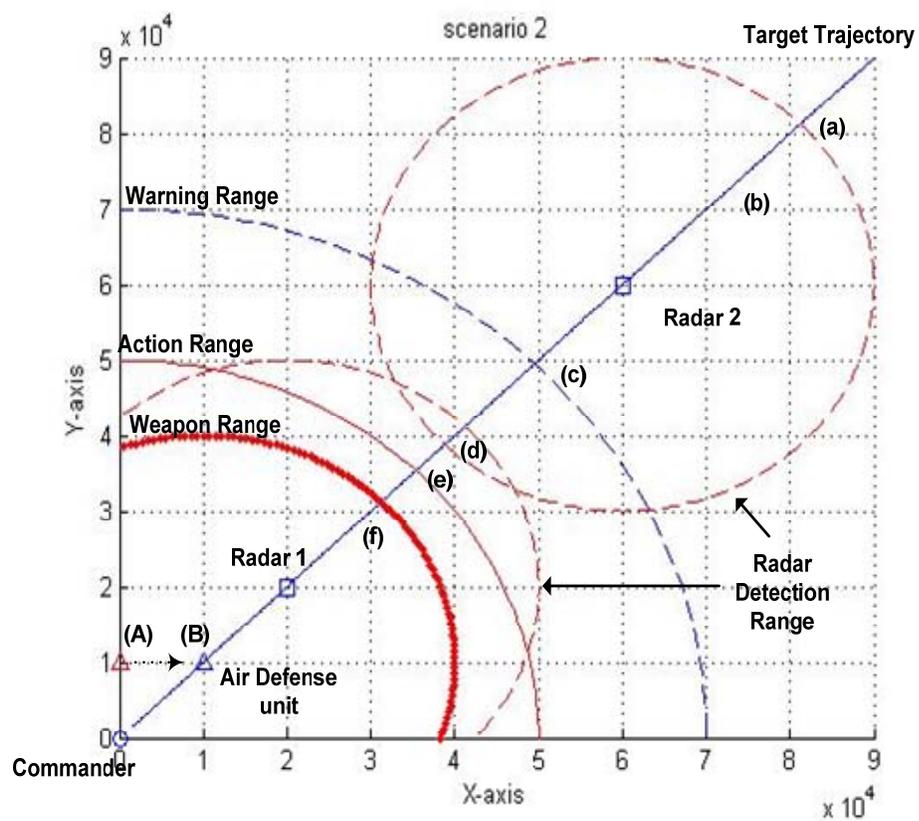


Figure 5-11 Illustration of scenario 2

## 5.3.2. DEVS models

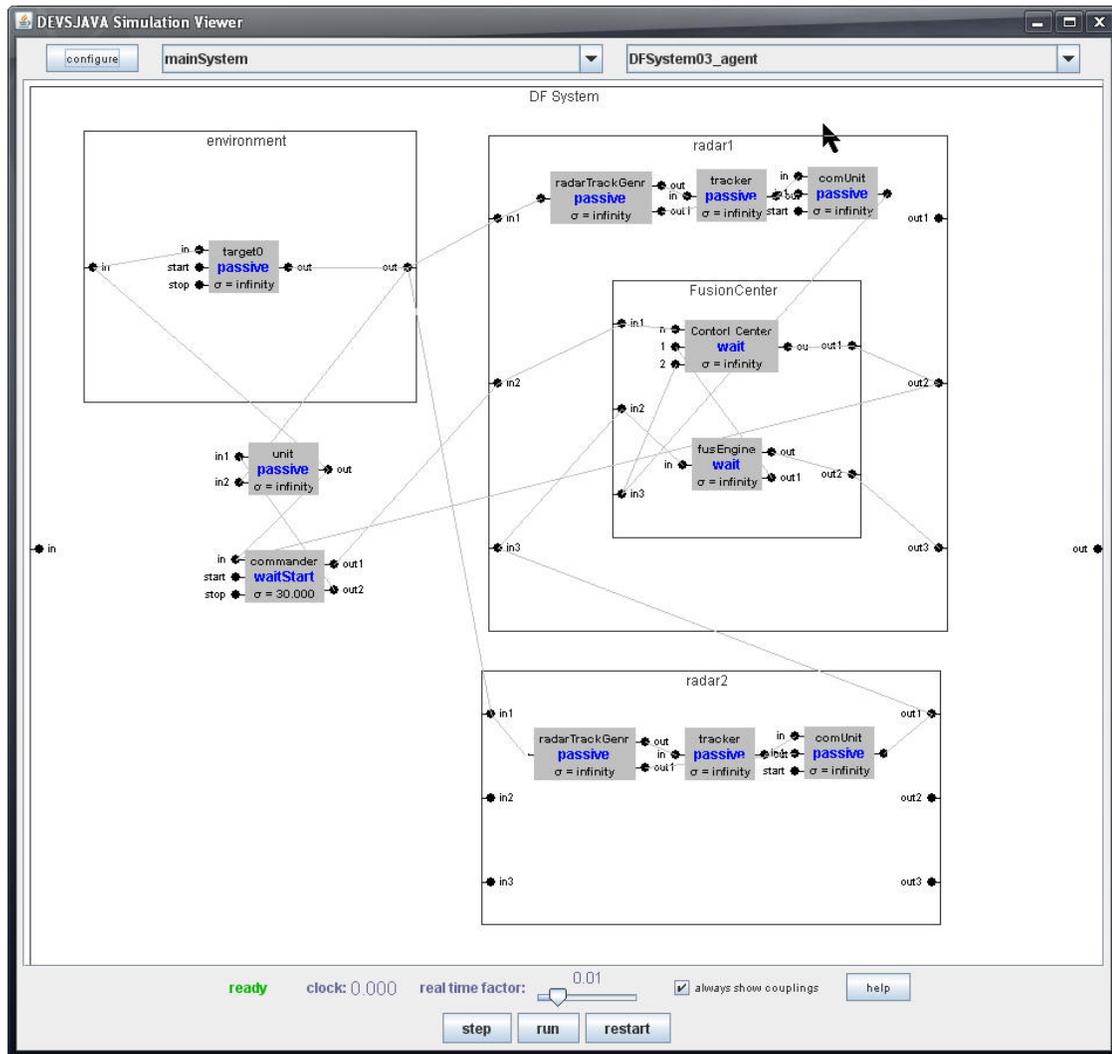


Figure 5-12 DEVS models for scenario 2 in Simview

We add three models for scenario 2: **unit**, **radar 2**, and **fusionEngine** in the **FusionCenter** coupled model. The **unit** is an agent model to perform air defense

operations according to the commander's orders. Figure 5-13 shows the model's behavior in the state diagram.

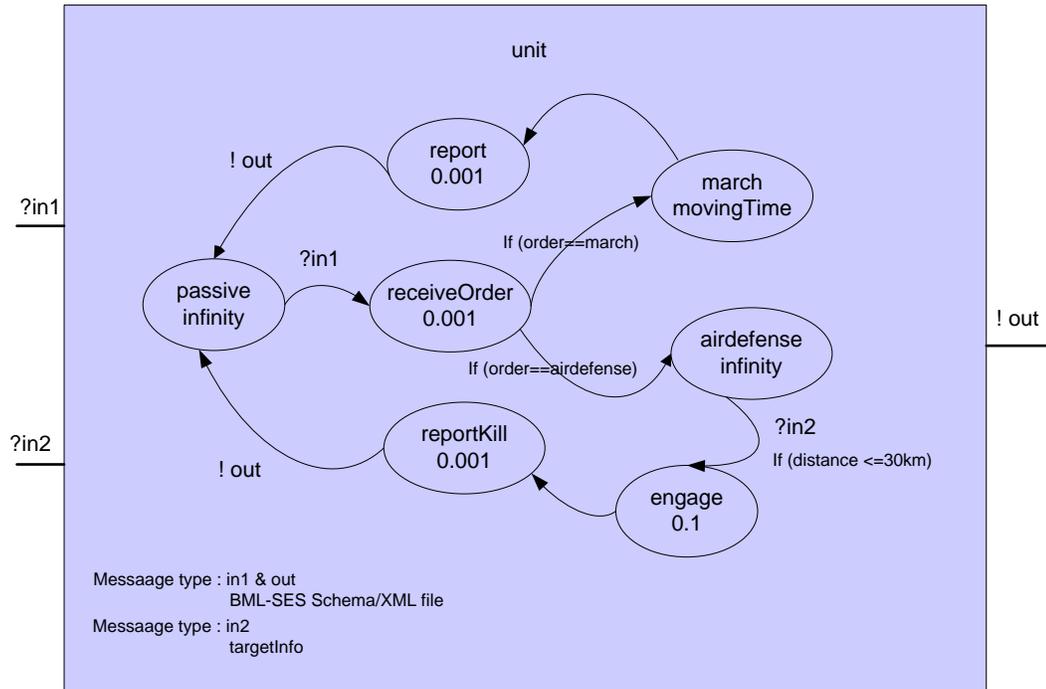


Figure 5-13 State diagram of **unit** model

The **unit** waits until it receives an order from the commander. First the order is to march to a specific location to carry out further operation in *movingTime* which is specified in the BML order. After moving to the location, it prepares for another operation and waits for the next order. The next order is for an air defense operation, since the target is a threat and we need to take care of the target immediately. If the target comes into Weapon Range boundary, the state moves to “engage.” We assume that the unit is equipped with high performance weapon and it can kill the target with the 90% probability. Then the unit engages until it kills the target in an automatic way.

The **radar2** is a pure radar model without interaction with the commander. It simply detects targets and generates track data of them. Then it sends the target data to the fusion center (**radar1**) in CoT message format.

The **fusionEngine** model performs fusion processes for multisensory tracking based on NN and gating techniques.

We also modify the **commander** model to issue orders and requests in Figure 5-14.

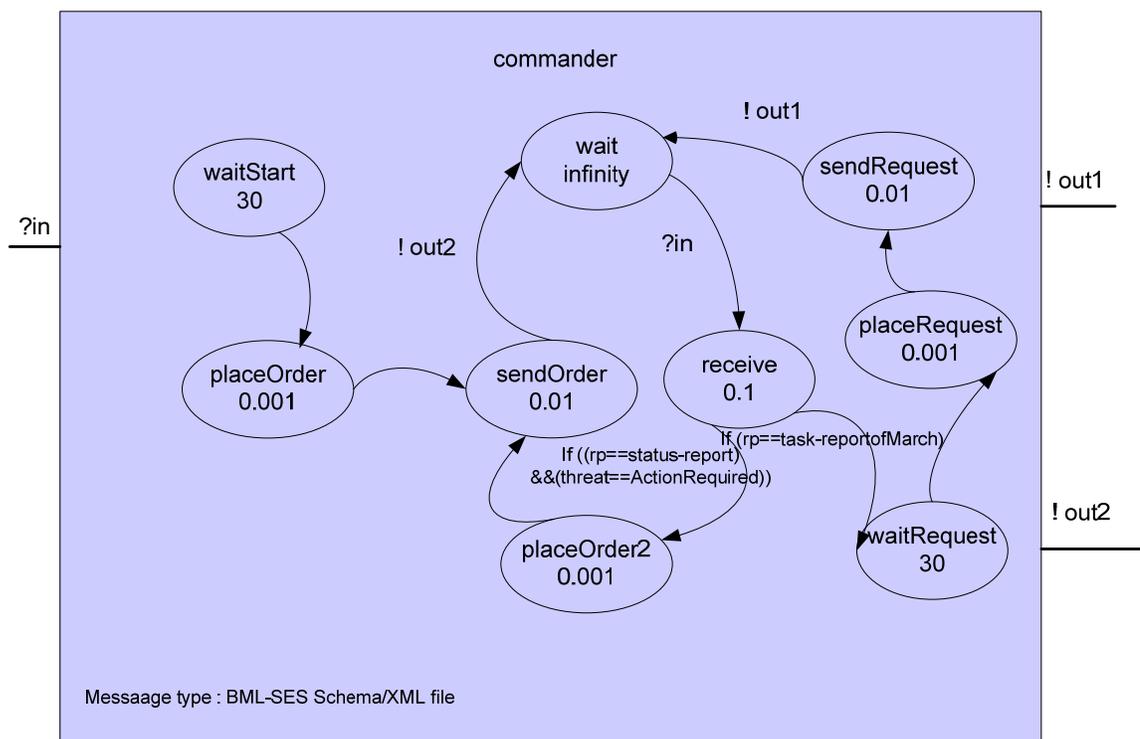


Figure 5-14 State diagram of **commander** model

Compared to scenario 1-1 or scenario 1-2's model, the **commander** model works more effectively. It issues orders to march and engage in accordance with the target's threat types. When the model receives a task-report from the **unit** model after giving an order to march, it places a request to receive information. When the **commander** model gets a status-report and the threat type is *ActionRequired*, it needs to act appropriately against the threat. As a result, the **commander** issues an order of airdefense to get the air defense unit engaged with the threat target when the **unit** is ready.

### 5.3.3. The results of simulation under scenario 2

The commander, located at the origin, issues an order to replace the air defense unit's position from (A) to (B) at which it can perform air defense operations effectively.

The results of relations and threat types at each state are listed in Table 5-6.

Target location	Relations	Threat type	Explanation
(a)	<i>NaN</i>	<i>NaN</i>	Target enters detection range of radar2
(b)	Fast,Closing,Neutral,Hostile, OutWRange,OutARange	Cautious	First response of a request
(c)	Fast,Closing,Neutral,Hostile, InWRange,OutARange	Threat	Comes in Warning Range
(d)	Fast, Closing,Neutral,Hostile, InWRange,OutARange	Threat	Radar1 takes over the target from radar2
(e)	Fast,Closing,Neutral,Hostile, InWRange,InARange	Action Required	Comes into Action Range

(f)	Fast,Closing,Neutral,Hostile, InWRange,InARange	Action Required	Comes into Weapon Range
-----	--	--------------------	----------------------------

Table 5-6 Changes of relations and threat types with respect to target location

From (a) radar 2 starts to generate target track data. The commander gives a request of Level 3 information and gets the first response from radar 1 (fusion center) at (b). As the target goes into the Warning Range, its threat type turns into *Threat* at (c). When the target passes over the cross-section of radar 1 and radar 2, the radar2 switches tracking authority to radar2 at (d). As the target comes across the Action Range boundary and gets close at (e), the commander issues an order to counter the threat. Finally, when it comes within the weapon's effective range, the unit, which has been taken the air defense order, engages the target and brings it down.

## CHAPTER 6. CONCLUSIONS AND FUTURE WORK

### 6.1. Conclusions

This dissertation is dedicated to introducing an efficient way to integrate DF systems with C2 systems to exchange information based on SES ontology in distributed data fusion networks. SES ontology expresses the information of a domain in a rigorous structural manner. It also facilitates message exchange in a network-centric environment. Message interchange is driven by pragmatic frames, which formalizes the types of requests that users wish to issue for information updates. BML is a formal command and control (C2) language used in the military realm, which states tasks, requests and reports in a clear way. Consequently, we can use BML as an interoperable interface for expressing pragmatic frames for commanders' requests of various levels of information in a military information exchange framework. A pragmatic frame in IEF is closely related to the user refinement level in the JDL model. Furthermore, BML enables human intervention into DF process so that it customizes the DF process in an efficient way.

Meanwhile, another interoperability issue for machine-to-machine communication is explored by investigating the possibility of using CoT in sensor networks. CoT is becoming popular in military communities because of its simple and open structure. The structured message type regulates three entities and 12 attributes, and the other part can be extended by users' objectives. Besides, it is uncomplicated work to implement the

XML-encoded message. As a result, we apply CoT as a message container to convey radar data to different radars over radar networks.

This dissertation's approach allows us to develop ontology-based data fusion processes within modeling and a simulation environment. The previous two proof-of-concept examples show how BML enables efficient exchange of information based on the SES ontology implemented in XML. The simulation-based evaluation shows the dynamic variations of relations between a commander and a target with respect to target states. The more detailed scenario shows the capability of the ontological DF scheme to support commanders' decisions in an automated way.

## 6.2. Future Work

This work is a conceptual study to explore the capability of a BML-driven Information Exchange Framework in SES. Therefore several follow-on studies remain. For one study, since our approach is an extension of existing BML, we need to investigate the proposed request system's compatibility with existing BML. Second, our examples show how we build up an air-battlefield situation. Therefore, generalization to fabricate the whole battlefield picture, including ground situation, should be considered. In addition, we need to consider the relation of our study in the Webservice SOA concept. Many information service applications such as Network Centric Enterprise Services (NCES) in the Global Information Grid (GIG) need request and answer interfaces which mediate bilateral conversations between humans and machines. The BML based information exchange



## APPENDIX I

### NATURAL LANGUAGE STYLE DESCRIPTION OF RADAR-SES

From the RadarSES perspective, RadarSES is made of Sensor, and Data!

From the Sensor perspective, Sensor is made of RadarID, Platform, and LocationInfo!

The RadarID has a radarid!

The range of RadarID's radarid is string!

The Platform has a type!

The range of Platform's type is string with values Airborn, Mounted, Towed, and FixedOnGround!

From the LocationInfo perspective, LocationInfo is made of X, and Y!

The X has a location!

The range of X's location is double!

The Y has a location!

The range of Y's location is double!

From the Data perspective, Data is made of TimeStamp, TrackID, Measurements, and IFF!

The TimeStamp has a sensingtime!

The range of TimeStamp's sensingtime is double!

The TrackID has a trkid!

The range of TrackID's trkid is string!

The IFF has a iffvalue!

The range of IFF's iffvalue is string with Hostile, Friend, UnIdentified, Pending!

A Measurements can be CartesianCoord, or PolarCoord in CoordinatingSystem!

From the CartesianCoord perspective, CartesianCoord is made of X\_tar, Y\_tar, X\_tar\_vel, Y\_tar\_vel, and Errors!

The X\_tar has a location!

The range of X\_tar's location is double!

The Y\_tar has a location!

The range of Y\_tar's location is double!

The X\_tar\_vel has velocity!

The range of X\_tar\_vel's velocity is double !

The Y\_tar\_vel has velocity!

The range of Y\_tar\_vel's velocity is double !

The Errors has a stateCov!

The range of Error's stateCov is matrix!

From the MathML perspective, Errors is made of matrix !

matrix has row, and column !

The range of matrix's row is int !

The range of matrix's column is int !

From the structure perspective, matrix is made of matrixrow !

From the structure perspective, matrixrow is made of cn !

cn has cellval !

The range of cn's cellvel is double !

From the PolarCoord perspective, PolarCoord is made of Bearing, Elevation, Range, and Errors!

The Bearing has a degree!

The range of Bearing's degree is double with values [0, 360]!

The Elevation has a degree!

The range of Elevation's degree is double with values [0, 360]!

The Range has a distance!

The range of Range's distance is double with values [0, 100000]!

The Errors has a degree!

## APPENDIX II

### RADAR-SES SCHEMA

```

<?xml version='1.0' encoding='us-ascii'?>

<xs:schema      xmlns:xs="http://www.w3.org/2001/XMLSchema"      elementFormDefault="qualified"
attributeFormDefault="unqualified">

  <xs:simpleType name="double0100000">
    <xs:restriction base="xs:double">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="100000"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="double0360">
    <xs:restriction base="xs:double">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="360"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="double0360">
    <xs:restriction base="xs:double">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="360"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="stringAirbornMountedTowedFixedOnGround">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Airborn"/>
      <xs:enumeration value="Mounted"/>
      <xs:enumeration value="Towed"/>
      <xs:enumeration value="FixedOnGround"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="IFF">
    <xs:complexType>
      <xs:attribute name="iffvalue" type="xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="TimeStamp">
    <xs:complexType>
      <xs:attribute name="sensingtime" type="xs:double" />
    </xs:complexType>
  </xs:element>

```

```

</xs:complexType>
</xs:element>

  <xs:element name="Range">
    <xs:complexType>
      <xs:attribute name="distance" type = "double0 100000" />
    </xs:complexType>
  </xs:element>

  <xs:element name="Elevation">
    <xs:complexType>
      <xs:attribute name="degree" type = "double0 360" />
    </xs:complexType>
  </xs:element>

  <xs:element name="Bearing">
    <xs:complexType>
      <xs:attribute name="degree" type = "double0 360" />
    </xs:complexType>
  </xs:element>

  <xs:element name="cn">
    <xs:complexType>
      <xs:attribute name="cellval" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "matrixrow-structureDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "cn"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="matrixrow">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfmatrixrow">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "matrixrow-structureDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name = "matrix-structureDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "matrixrow"/>

```

```

</xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="matrix">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfmatrix">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "matrix-structureDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="column" type = "xs:int" />
      <xs:attribute name="row" type = "xs:int" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "Errors-MathMLDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "matrix"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Errors">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfErrors">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "Errors-MathMLDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="degree" />
      <xs:attribute name="stateCov" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "PolarCoord-PolarCoordDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "Range"/>
        <xs:element ref = "Elevation"/>
        <xs:element ref = "Bearing"/>
        <xs:element ref = "Errors"/>
      </xs:sequence>
    </xs:complexType>

```

```

</xs:element>

  <xs:element name="PolarCoord">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfPolarCoord">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "PolarCoord-PolarCoordDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Y_tar_vel">
    <xs:complexType>
      <xs:attribute name="velocity" type = "xs:double" />
    </xs:complexType>
  </xs:element>

  <xs:element name="Y_tar">
    <xs:complexType>
      <xs:attribute name="location" type = "xs:double" />
    </xs:complexType>
  </xs:element>

  <xs:element name="X_tar">
    <xs:complexType>
      <xs:attribute name="location" type = "xs:double" />
    </xs:complexType>
  </xs:element>

  <xs:element name="X_tar_vel">
    <xs:complexType>
      <xs:attribute name="velocity" type = "xs:double" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "CartesianCoord-CartesianCoordDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "Y_tar_vel"/>
        <xs:element ref = "Y_tar"/>
        <xs:element ref = "X_tar"/><xs:element ref="Errors"/>
        <xs:element ref = "X_tar_vel"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="CartesianCoord">
    <xs:complexType>

```

```

    <xs:sequence>
    <xs:element name = "aspectsOfCartesianCoord">
      <xs:complexType>
        <xs:choice>
          <xs:element ref = "CartesianCoord-CartesianCoordDec"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name = "Measurements-CoordinatingSystemSpec">
    <xs:complexType>
      <xs:choice>
        <xs:element ref = "PolarCoord"/>
        <xs:element ref = "CartesianCoord"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>

  <xs:element name="Measurements">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "Measurements-CoordinatingSystemSpec"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="TrackID">
    <xs:complexType>
      <xs:attribute name="trkid" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "Data-DataDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "IFF"/>
        <xs:element ref = "TimeStamp"/>
        <xs:element ref = "Measurements"/>
        <xs:element ref = "TrackID"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Data">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfData">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "Data-DataDec"/>

```

```

</xs:choice>
  </xs:complexType>
</xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="RadarID">
    <xs:complexType>
<xs:attribute name="radarid" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="Y">
    <xs:complexType>
<xs:attribute name="location" type = "xs:double" />
    </xs:complexType>
  </xs:element>

  <xs:element name="X">
    <xs:complexType>
<xs:attribute name="location" type = "xs:double" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "LocationInfo-LocationInfoDec">
    <xs:complexType>
<xs:sequence>
  <xs:element ref = "Y"/>
  <xs:element ref = "X"/>
</xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="LocationInfo">
    <xs:complexType>
  <xs:sequence>
<xs:element name = "aspectsOfLocationInfo">
  <xs:complexType>
<xs:choice>
  <xs:element ref = "LocationInfo-LocationInfoDec"/>
</xs:choice>
  </xs:complexType>
</xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="Platform">
    <xs:complexType>
<xs:attribute name="type" type = "stringAirbornMountedTowedFixedOnGround" />
    </xs:complexType>
  </xs:element>

```

```

    <xs:element name = "Sensor-SensorDec">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "RadarID"/>
    <xs:element ref = "LocationInfo"/>
    <xs:element ref = "Platform"/>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="Sensor">
    <xs:complexType>
    <xs:sequence>
    <xs:element name = "aspectsOfSensor">
    <xs:complexType>
    <xs:choice>
    <xs:element ref = "Sensor-SensorDec"/>
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name = "RadarSES-RadarSESDec">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "Data"/>
    <xs:element ref = "Sensor"/>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="RadarSES">
    <xs:complexType>
    <xs:sequence>
    <xs:element name = "aspectsOfRadarSES">
    <xs:complexType>
    <xs:choice>
    <xs:element ref = "RadarSES-RadarSESDec"/>
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    </xs:schema>

```

## APPENDIX III

## PRUNED RADAR-SES IN XML DOCUMENT FORMAT

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<RadarSES>
  <RadarSES-RadarSESDec>
    <Data>
      <Data-DataDec>
        <IFF type="string">Hostile</IFF>
        <TimeStamp type="double">170.0</TimeStamp>
        <Measurements>
          <Measurements-CoordinatingSystemSpec>
            <CartesianCoord>
              <CartesianCoord-CartesianCoordDec>
                <Y_tar_vel type="double">-225.55529255579322</Y_tar_vel>
                <Y_tar type="double">51480.68904834744</Y_tar>
                <X_tar type="double">72037.87624776001</X_tar>
              <Errors>
                <Errors-MathMLDec>
                  <matrix type="int">
                    <matrix-structureDec>
                      <matrixrow>
                        <matrixrow-structureDec>
                          <cn>24.96820844083777</cn>
                          <cn>0.0</cn>
                          <cn>6.303923298307382</cn>
                          <cn>0.0</cn>
                        </matrixrow-structureDec>
                      </matrixrow>
                      <matrixrow>
                        <matrixrow-structureDec>
                          <cn>0.0</cn>
                          <cn>24.96820844083777</cn>
                          <cn>0.0</cn>
                          <cn>6.303923298307382</cn>
                        </matrixrow-structureDec>
                      </matrixrow>
                      <matrixrow>
                        <matrixrow-structureDec>
                          <cn>6.303923298304653</cn>
                          <cn>0.0</cn>
                          <cn>365.18528856254693</cn>
                          <cn>0.0</cn>
                        </matrixrow-structureDec>
                      </matrixrow>
                    </matrix-structureDec>
                  </matrixrow>
                </matrixrow>
              </Errors-MathMLDec>
            </Errors>
          </Measurements-CoordinatingSystemSpec>
        </Measurements>
      </Data-DataDec>
    </Data>
  </RadarSES-RadarSESDec>
</RadarSES>

```

```

        <matrixrow-structureDec>
            <cn>0.0</cn>
            <cn>6.303923298304653</cn>
            <cn>0.0</cn>
            <cn>365.18528856254693</cn>
        </matrixrow-structureDec>
    </matrixrow>
</matrix-structureDec>
</matrix>
</Errors-MathMLDec>
</Errors>
    <X_tar_vel type="double">-109.30073947329835</X_tar_vel>
</CartesianCoord-CartesianCoordDec>
</CartesianCoord>
</Measurements-CoordinatingSystemSpec>
</Measurements>
    <TrackID type="string">r001-0001</TrackID>
</Data-DataDec>
</Data>
<Sensor>
    <Sensor-SensorDec>
        <RadarID type="string">r001</RadarID>
        <LocationInfo>
            <LocationInfo-LocationInfoDec>
                <Y type="double">30000.0</Y>
                <X type="double">30000.0</X>
            </LocationInfo-LocationInfoDec>
        </LocationInfo>
        <Platform type="string">fixedOnGround</Platform>
    </Sensor-SensorDec>
</Sensor>
</RadarSES-RadarSESDec>
</RadarSES>

```

## APPENDIX IV

### BML-SES SCHEMA

```
<?xml version='1.0' encoding='us-ascii'?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
```

```
<xs:simpleType name="double01000">
<xs:restriction base="xs:double">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="1000"/>
</xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="double010000">
<xs:restriction base="xs:double">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="10000"/>
</xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="double010000">
<xs:restriction base="xs:double">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="10000"/>
</xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="double010000">
<xs:restriction base="xs:double">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="10000"/>
</xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="double010000">
<xs:restriction base="xs:double">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="10000"/>
</xs:restriction>
</xs:simpleType>
```

```
<xs:simpleType name="double010000">
<xs:restriction base="xs:double">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="10000"/>
```

```

</xs:restriction>
</xs:simpleType>

<xs:simpleType name="double010000">
<xs:restriction base="xs:double">
<xs:minInclusive value="0"/>
<xs:maxInclusive value="10000"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="stringfactplausibleuncertainindeterminate">
<xs:restriction base="xs:string">
<xs:enumeration value="fact"/>
<xs:enumeration value="plausible"/>
<xs:enumeration value="uncertain"/>
<xs:enumeration value="indeterminate"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="stringfactplausibleuncertainindeterminate">
<xs:restriction base="xs:string">
<xs:enumeration value="fact"/>
<xs:enumeration value="plausible"/>
<xs:enumeration value="uncertain"/>
<xs:enumeration value="indeterminate"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="stringfactplausibleuncertainindeterminate">
<xs:restriction base="xs:string">
<xs:enumeration value="fact"/>
<xs:enumeration value="plausible"/>
<xs:enumeration value="uncertain"/>
<xs:enumeration value="indeterminate"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="stringFriendHostileUnknown">
<xs:restriction base="xs:string">
<xs:enumeration value="Friend"/>
<xs:enumeration value="Hostile"/>
<xs:enumeration value="Unknown"/>
</xs:restriction>
</xs:simpleType>

  <xs:element name="TargetYs">
    <xs:complexType>
<xs:attribute name="speed" type = "xs:double" />
    </xs:complexType>
  </xs:element>

  <xs:element name="Radius">
<xs:complexType>

```

```

<xs:attribute name="length" type = "double0 1000" />
</xs:complexType>
</xs:element>

  <xs:element name="TargetY">
<xs:complexType>
<xs:attribute name="locationY" type = "double010000" />
</xs:complexType>
</xs:element>

  <xs:element name="TargetXs">
<xs:complexType>
<xs:attribute name="speed" type = "xs:double" />
</xs:complexType>
</xs:element>

  <xs:element name="TargetX">
<xs:complexType>
<xs:attribute name="locationX" type = "double010000" />
</xs:complexType>
</xs:element>

  <xs:element name = "TargetWhere-contentDec">
<xs:complexType>
<xs:sequence>
  <xs:element ref = "TargetYs"/>
  <xs:element ref = "Radius"/>
  <xs:element ref = "TargetY"/>
  <xs:element ref = "TargetXs"/>
  <xs:element ref = "TargetX"/>
</xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="TargetWhere">
<xs:complexType>
<xs:sequence>
<xs:element name = "aspectsOfTargetWhere">
  <xs:complexType>
<xs:choice>
  <xs:element ref = "TargetWhere-contentDec"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="TaskeeY">
<xs:complexType>
<xs:attribute name="location" type = "double010000" />
</xs:complexType>
</xs:element>

```

```

    <xs:element name="TaskeeX">
    <xs:complexType>
    <xs:attribute name="location" type = "double010000" />
    </xs:complexType>
    </xs:element>

    <xs:element name = "TaskeeWhere-contentDec">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "TaskeeY"/>
    <xs:element ref = "TaskeeX"/>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="TaskeeWhere">
    <xs:complexType>
    <xs:sequence>
    <xs:element name = "aspectsOfTaskeeWhere">
    <xs:complexType>
    <xs:choice>
    <xs:element ref = "TaskeeWhere-contentDec"/>
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="TaskerX">
    <xs:complexType>
    <xs:attribute name="txlocation" type = "double010000" />
    </xs:complexType>
    </xs:element>

    <xs:element name="TaskerY">
    <xs:complexType>
    <xs:attribute name="tylocation" type = "double010000" />
    </xs:complexType>
    </xs:element>

    <xs:element name = "TaskerWhere-contentDec">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "TaskerX"/>
    <xs:element ref = "TaskerY"/>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="TaskerWhere">
    <xs:complexType>

```

```

    <xs:sequence>
    <xs:element name = "aspectsOfTaskerWhere">
      <xs:complexType>
        <xs:choice>
          <xs:element ref = "TaskerWhere-contentDec"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

  <xs:element name = "WHERE-contentDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "TargetWhere"/>
        <xs:element ref = "TaskerWhere"/>
        <xs:element ref = "TaskerWhere"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="WHERE">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfWHERE">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "WHERE-contentDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="status-report">
    <xs:complexType>
      <xs:attribute name="certainty" type = "stringfactplausibleuncertainindeterminate" />
      <xs:attribute name="label" type = "xs:string" />
      <xs:attribute name="regarding" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="event-report">
    <xs:complexType>
      <xs:attribute name="everb" type = "xs:string" />
      <xs:attribute name="certainty" type = "stringfactplausibleuncertainindeterminate" />
      <xs:attribute name="label" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="task-report">

```

```

<xs:complexType>
<xs:attribute name="certainty" type = "stringfactplausibleuncertainindeterminate" />
<xs:attribute name="label" type = "xs:string" />
</xs:complexType>
</xs:element>

<xs:element name = "Report-sortDec">
<xs:complexType>
<xs:sequence>
<xs:element ref = "status-report"/>
<xs:element ref = "event-report"/>
<xs:element ref = "task-report"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="Report">
<xs:complexType>
<xs:sequence>
<xs:element name = "aspectsOfReport">
<xs:complexType>
<xs:choice>
<xs:element ref = "Report-sortDec"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="Order">
<xs:complexType>
<xs:attribute name="verb" type = "xs:string" />
<xs:attribute name="label" type = "xs:string" />
</xs:complexType>
</xs:element>

<xs:element name="Level2">
<xs:complexType>
<xs:attribute name="label" type = "xs:string" />
</xs:complexType>
</xs:element>

<xs:element name="Level3">
<xs:complexType>
<xs:attribute name="label" type = "xs:string" />
</xs:complexType>
</xs:element>

<xs:element name="Level1">
<xs:complexType>
<xs:attribute name="label" type = "xs:string" />
</xs:complexType>

```

```

</xs:element>

<xs:element name = "Request-FusionLevelDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref = "Level2"/>
      <xs:element ref = "Level3"/>
      <xs:element ref = "Level1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

  <xs:element name="Request">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfRequest">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "Request-FusionLevelDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name = "WHAT-FunctionSpec">
    <xs:complexType>
      <xs:choice>
        <xs:element ref = "Report"/>
        <xs:element ref = "Order"/>
        <xs:element ref = "Request"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>

  <xs:element name="WHAT">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "WHAT-FunctionSpec"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="WHY">
    <xs:complexType>
      <xs:attribute name="content" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="Interval">
    <xs:complexType>
      <xs:attribute name="intervalTime" type = "xs:string" />

```

```

</xs:complexType>
</xs:element>

  <xs:element name="Start">
<xs:complexType>
<xs:attribute name="startTime" type = "xs:string" />
</xs:complexType>
</xs:element>

  <xs:element name="Stop">
<xs:complexType>
<xs:attribute name="stopTime" type = "xs:string" />
</xs:complexType>
</xs:element>

  <xs:element name = "WHEN-contentDec">
<xs:complexType>
<xs:sequence>
  <xs:element ref = "Interval"/>
  <xs:element ref = "Start"/>
  <xs:element ref = "Stop"/>
</xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="WHEN">
<xs:complexType>
<xs:sequence>
<xs:element name = "aspectsOfWHEN">
  <xs:complexType>
<xs:choice>
  <xs:element ref = "WHEN-contentDec"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="AsSituation">
<xs:complexType>
<xs:attribute name="relations" type = "xs:string" />
</xs:complexType>
</xs:element>

  <xs:element name="Affiliation">
<xs:complexType>
<xs:attribute name="iffvalue" type = "stringFriendHostileUnknown" />
</xs:complexType>
</xs:element>

  <xs:element name="TrackID">
<xs:complexType>

```

```

<xs:attribute name="tID" type = "xs:string" />
</xs:complexType>
</xs:element>

<xs:element name = "AsObject-contentDec">
<xs:complexType>
<xs:sequence>
<xs:element ref = "Affiliation"/>
<xs:element ref = "TrackID"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="AsObject">
<xs:complexType>
<xs:sequence>
<xs:element name = "aspectsOfAsObject">
<xs:complexType>
<xs:choice>
<xs:element ref = "AsObject-contentDec"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="AsThreat">
<xs:complexType>
<xs:attribute name="threats" type = "xs:string" />
</xs:complexType>
</xs:element>

<xs:element name = "TargetWho-contentDec">
<xs:complexType>
<xs:sequence>
<xs:element ref = "AsSituation"/>
<xs:element ref = "AsObject"/>
<xs:element ref = "AsThreat"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="TargetWho">
<xs:complexType>
<xs:sequence>
<xs:element name = "aspectsOfTargetWho">
<xs:complexType>
<xs:choice>
<xs:element ref = "TargetWho-contentDec"/>
</xs:choice>
</xs:complexType>
</xs:element>

```

```

</xs:sequence>
</xs:complexType>
</xs:element>

  <xs:element name="TaskeeWho">
    <xs:complexType>
      <xs:attribute name="taskeeID" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name="TaskerWho">
    <xs:complexType>
      <xs:attribute name="taskerID" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "WHO-contentDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "TargetWho"/>
        <xs:element ref = "TaskeeWho"/>
        <xs:element ref = "TaskerWho"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="WHO">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfWHO">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "WHO-contentDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name = "BML-componentIDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "WHERE"/>
        <xs:element ref = "WHAT"/>
        <xs:element ref = "WHY"/>
        <xs:element ref = "WHEN"/>
        <xs:element ref = "WHO"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="BML">

```

```
<xs:complexType>
  <xs:sequence>
    <xs:element name = "aspectsOfBML">
      <xs:complexType>
        <xs:choice>
          <xs:element ref = "BML-componentIDec"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## APPENDIX V

### COT 2.0-SES SCHEMA

```

<?xml version='1.0' encoding='us-ascii'?>

<xs:schema      xmlns:xs="http://www.w3.org/2001/XMLSchema"      elementFormDefault="qualified"
attributeFormDefault="unqualified">

    <xs:element name="point">
        <xs:complexType>
            <xs:attribute name="lon" type = "decimal" />
            <xs:attribute name="le" type = "decimal" />
            <xs:attribute name="hae" type = "decimal" />
            <xs:attribute name="ce" type = "decimal" />
            <xs:attribute name="lat" type = "decimal" />
        </xs:complexType>
    </xs:element>

    <xs:element name = "event-contentDec">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref = "point"/><xs:element name = "detail"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="event">
        <xs:complexType>
            <xs:sequence>
                <xs:element name = "aspectsOfevent">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element ref = "event-contentDec"/>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="uid" type = "xs:string" />
            <xs:attribute name="time" type = "dateTime" />
            <xs:attribute name="how" type = "xs:string" />
            <xs:attribute name="start" type = "dateTime" />
            <xs:attribute name="type" type = "xs:string" />
            <xs:attribute name="stale" type = "dateTime" />
            <xs:attribute name="version" />
        </xs:complexType>
    </xs:element>

```

```
<xs:element name = "CoT2.0-SES-structureDec">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref = "event"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

  <xs:element name="CoT2.0-SES">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfCoT2.0-SES">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "CoT2.0-SES-structureDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

## APPENDIX VI

### COT 3.0-SES SCHEMA

```

<?xml version='1.0' encoding='us-ascii'?>

<xs:schema      xmlns:xs="http://www.w3.org/2001/XMLSchema"      elementFormDefault="qualified"
attributeFormDefault="unqualified">

    <xs:element name="refevent">
    <xs:complexType>
    <xs:attribute name="diffrg:id" type = "xs:string" />
    </xs:complexType>
    </xs:element>

    <xs:element name = "allevents-structureMultiAsp">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "refevent" minOccurs= "0" maxOccurs= "10"/>
    </xs:sequence>
    <xs:attribute name="numContainedInallevents" type="xs:int"></xs:attribute>
    </xs:complexType>
    </xs:element>

    <xs:element name="allevents">
    <xs:complexType>
    <xs:sequence>
    <xs:element name = "aspectsOfallevents">
    <xs:complexType>
    <xs:choice>
    <xs:element ref = "allevents-structureMultiAsp"/>
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="refpoint">
    <xs:complexType>
    <xs:attribute name="diffrg:id" type = "xs:string" />
    </xs:complexType>
    </xs:element>

    <xs:element name = "allpoints-structureMultiAsp">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "refpoint" minOccurs= "0" maxOccurs= "10"/>

```

```

</xs:sequence>
  <xs:attribute name="numContainedInallpoints" type="xs:int"></xs:attribute>
</xs:complexType>
</xs:element>

  <xs:element name="allpoints">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfallpoints">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "allpoints-structureMultiAsp"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name = "detail-referenceDec">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "allevents"/>
        <xs:element ref = "allpoints"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="detail">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfdetail">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "detail-referenceDec"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="elementOrder" type = "xs:int" />
      <xs:attribute name="diffrg:id" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "details-contentMultiAsp">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "detail" minOccurs= "0" maxOccurs= "10"/>
      </xs:sequence>
      <xs:attribute name="numContainedIndetails" type="xs:int"></xs:attribute>
    </xs:complexType>
  </xs:element>

```

```

    <xs:element name="details">
    <xs:complexType>
    <xs:sequence>
    <xs:element name = "aspectsOfdetails">
    <xs:complexType>
    <xs:choice>
    <xs:element ref = "details-contentMultiAsp"/>
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="refdetail">
    <xs:complexType>
    <xs:attribute name="diffrg:id" type = "xs:string" />
    </xs:complexType>
    </xs:element>

    <xs:element name = "alldetails-structureMultiAsp">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "refdetail" minOccurs= "0" maxOccurs= "10"/>
    </xs:sequence>
    <xs:attribute name="numContainedInalldetails" type="xs:int"></xs:attribute>
    </xs:complexType>
    </xs:element>

    <xs:element name="alldetails">
    <xs:complexType>
    <xs:sequence>
    <xs:element name = "aspectsOfalldetails">
    <xs:complexType>
    <xs:choice>
    <xs:element ref = "alldetails-structureMultiAsp"/>
    </xs:choice>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name = "event-referenceDec">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref = "alldetails"/><xs:element ref="allpoints"/>
    </xs:sequence>
    </xs:complexType>
    </xs:element>

    <xs:element name="event">
    <xs:complexType>

```

```

    <xs:sequence>
    <xs:element name = "aspectsOfevent">
      <xs:complexType>
        <xs:choice>
          <xs:element ref = "event-referenceDec"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="uid" type = "xs:string" />
  <xs:attribute name="elementOrder" type = "xs:int" />
  <xs:attribute name="time" type = "dateTime" />
  <xs:attribute name="qos" type = "xs:string" />
  <xs:attribute name="how" type = "xs:string" />
  <xs:attribute name="type" type = "xs:string" />
  <xs:attribute name="stale" type = "dateTime" />
  <xs:attribute name="diffrg:id" type = "xs:string" />
  </xs:complexType>
</xs:element>

  <xs:element name = "events-contentMultiAsp">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref = "event" minOccurs= "0" maxOccurs= "10"/>
      </xs:sequence>
      <xs:attribute name="numContainedInevents" type="xs:int"/></xs:attribute>
    </xs:complexType>
  </xs:element>

  <xs:element name="events">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "aspectsOfevents">
          <xs:complexType>
            <xs:choice>
              <xs:element ref = "events-contentMultiAsp"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="access" type = "xs:string" />
      <xs:attribute name="opex" type = "xs:string" />
      <xs:attribute name="version" type = "xs:string" />
    </xs:complexType>
  </xs:element>

  <xs:element name = "point-referenceDec">
    <xs:complexType>
      <xs:sequence><xs:element ref="allevents"/><xs:element ref="alldetails"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    <xs:element name="point">
      <xs:complexType>
        <xs:sequence>
          <xs:element name = "aspectsOfpoint">
            <xs:complexType>
              <xs:choice>
                <xs:element ref = "point-referenceDec"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="elementOrder" type = "xs:int" />
        <xs:attribute name="lon" type = "decimal" />
        <xs:attribute name="le" type = "decimal" />
        <xs:attribute name="hae" type = "decimal" />
        <xs:attribute name="ce" type = "decimal" />
        <xs:attribute name="lat" type = "decimal" />
        <xs:attribute name="diffrg:id" type = "xs:string" />
      </xs:complexType>
    </xs:element>

    <xs:element name = "points-contentMultiAsp">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref = "point" minOccurs= "0" maxOccurs= "10"/>
        </xs:sequence>
        <xs:attribute name="numContainedInpoints" type="xs:int"></xs:attribute>
      </xs:complexType>
    </xs:element>

    <xs:element name="points">
      <xs:complexType>
        <xs:sequence>
          <xs:element name = "aspectsOfpoints">
            <xs:complexType>
              <xs:choice>
                <xs:element ref = "points-contentMultiAsp"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

    <xs:element name = "CoT3.0-SES-structureDec">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref = "details"/>
          <xs:element ref = "events"/>
          <xs:element ref = "points"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

```
<xs:element name="CoT3.0-SES">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "aspectsOfCoT3.0-SES">
        <xs:complexType>
          <xs:choice>
            <xs:element ref = "CoT3.0-SES-structureDec"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

## APPENDIX VII

### COT 3.0 XML FILE WITH RADAR DATA

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<diffgr:diffgram xmlns:diffgr="urn:schemas-mitre-com:xml-diffgram-v1" xmlns:mitredata="urn:schemas-
mitre-org:xml-mitredata" xmlns:xsd="http://www.w3.org/2002/XMLSchema">
  <CoT3.0-SES>
    <CoT3.0-SES-structureDec>
      <details>
        <details-contentMultiAsp>
          <detail diffgr:id="detail01" elementOrder="1">
            <RadarSES>
<RadarSES-RadarSESDec>
  <Data>
    <Data-DataDec>
      <IFF type="string">Hostile</IFF>
      <TimeStamp type="double">140.0</TimeStamp>
      <Measurements>
        <Measurements-CoordinatingSystemSpec>
          <CartesianCoord>
            <CartesianCoord-CartesianCoordDec>
              <Y_tar_vel type="double">-223.25163131017158</Y_tar_vel>
              <Y_tar type="double">58287.14925838465</Y_tar>
              <X_tar type="double">75211.2204871975</X_tar>
            <Errors>
              <Errors-MathMLDec>
                <matrix type="int">
                  <matrix-structureDec>
                    <matrixrow>
                      <matrixrow-structureDec>
                        <cn>24.9682087123947</cn>
                        <cn>0.0</cn>
                        <cn>6.303912373887215</cn>
                        <cn>0.0</cn>
                      </matrixrow-structureDec>
                    </matrixrow>
                    <matrixrow>
                      <matrixrow-structureDec>
                        <cn>0.0</cn>
                        <cn>24.9682087123947</cn>
                        <cn>0.0</cn>
                        <cn>6.303912373887215</cn>
                      </matrixrow-structureDec>
                    </matrixrow>
                    <matrixrow>
                      <matrixrow-structureDec>
                        <cn>6.303912373887215</cn>
                      </matrixrow-structureDec>
                    </matrixrow>
                  </matrix>
            </Errors>
          </Measurements-CoordinatingSystemSpec>
        </Measurements>
      </Data-DataDec>
    </Data>
  </RadarSES-RadarSESDec>
</RadarSES>
</detail>
</details-contentMultiAsp>
</details>
</CoT3.0-SES-structureDec>
</CoT3.0-SES>

```

```

        <cn>0.0</cn>
        <cn>365.1857280405652</cn>
        <cn>0.0</cn>
      </matrixrow-structureDec>
    </matrixrow>
    <matrixrow>
      <matrixrow-structureDec>
        <cn>0.0</cn>
        <cn>6.303912373887215</cn>
        <cn>0.0</cn>
        <cn>365.1857280405652</cn>
      </matrixrow-structureDec>
    </matrixrow>
  </matrix-structureDec>
</matrix>
</Errors-MathMLDec>
</Errors>
  <X_tar_vel type="double">-105.80908428470397</X_tar_vel>
</CartesianCoord-CartesianCoordDec>
</CartesianCoord>
</Measurements-CoordinatingSystemSpec>
</Measurements>
  <TrackID type="string">r001-0001</TrackID>
</Data-DataDec>
</Data>
<Sensor>
  <Sensor-SensorDec>
    <RadarID type="string">r001</RadarID>
    <LocationInfo>
      <LocationInfo-LocationInfoDec>
        <Y type="double">30000.0</Y>
        <X type="double">30000.0</X>
      </LocationInfo-LocationInfoDec>
    </LocationInfo>
    <Platform type="string">fixedOnGround</Platform>
  </Sensor-SensorDec>
</Sensor>
</RadarSES-RadarSESDec>
</RadarSES>
  <detail-referenceDec>
    <allevents>
      <allevents-structureMultiAsp>
        <refevent diffrg:id="event01"/>
      </allevents-structureMultiAsp>
    </allevents>
    <allpoints>
      <allpoints-structureMultiAsp>
        <refpoint diffrg:id="point01"/>
      </allpoints-structureMultiAsp>
    </allpoints>
  </detail-referenceDec>
</detail>
</details-contentMultiAsp>

```

```

</details>
<events version="3.0">
  <events-contentMultiAsp>
    <event diffrg:hasChanges="modified" diffrg:id="event01" elementOrder="1" how="m-
p" stale="1" time="140.0" type="a-h-A" uid="r001-0001">
      <event-referenceDec>
        <alldetails>
          <alldetails-structureMultiAsp>
            <refdetail diffrg:id="detail01"/>
          </alldetails-structureMultiAsp>
        </alldetails>
        <allpoints>
          <allpoints-structureMultiAsp>
            <refpoint diffrg:id="point01"/>
          </allpoints-structureMultiAsp>
        </allpoints>
      </event-referenceDec>
    </event>
  </events-contentMultiAsp>
</events>
<points>
  <points-contentMultiAsp>
    <point ce="10" diffrg:id="point01" elementOrder="1" hae="-6282035.228819944"
lat="0.1868433575685564" le="1" lon="0.6592972209257126">
      <point-referenceDec>
        <allevents>
          <allevents-structureMultiAsp>
            <refevent diffrg:id="event01"/>
          </allevents-structureMultiAsp>
        </allevents>
        <alldetails>
          <alldetails-structureMultiAsp>
            <refdetail diffrg:id="detail01"/>
          </alldetails-structureMultiAsp>
        </alldetails>
      </point-referenceDec>
    </point>
  </points-contentMultiAsp>
</points>
</CoT3.0-SES-structureDec>
</CoT3.0-SES>
</diffgr:diffgram>

```

## APPENDIX VIII

### DATUM CONVERSION FOR TRACKDISPLAY [67].

The Cartesian coordinating system is called Earth-Centered Earth-Fixed (ECEF). OpenMap uses more common geodetic-mapping coordinates of Latitude, Longitude, and Altitude (LLA). Geodetic coordinates can be converted to a second map reference known as Mercator Projections including Universal Transverse Mercator (UTM) or USGS Grid system.

As a preliminary step, I set up some parameters of WGS84 to use in the conversions since its origin is coincident with the ECEF origin.

A reference ellipsoid can be described by several parameters that define its shape as follows:

- WGS84 Parameters

$$a = 6378137 \quad (\text{Equation A-7-1})$$

$$b = a(1 - f) \quad (\text{Equation A-7-2})$$

$$f = \frac{1}{298.257223563} \quad (\text{Equation A-7-3})$$

$$e = \sqrt{\frac{a^2 - b^2}{a^2}} \quad (\text{Equation A-7-4})$$

$$e' = \sqrt{\frac{a^2 - b^2}{b^2}} \quad (\text{Equation A-7-5})$$

Now I define the following coordinates variables:

- ECEF :  $(x, y, z)$

where  $x$  is a point in X-axis,  $y$  is a point in Y-axis, and  $z$  is a point in Z-axis

- LLA :  $(\varphi, \lambda, h)$

where  $\varphi$  denotes latitude,  $\lambda$  denotes longitude, and  $h$  is height.

The relations is shown in Figure A-7-1.

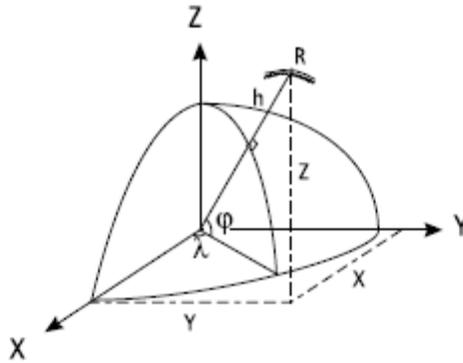


Figure A-7-1 Relations between ECEF and LLA

### 1. From LLA to ECEF

One conversion method is addressed.

$$x = (N + h) \cos \varphi \cos \lambda \quad (\text{Equation A-7-6})$$

$$y = (N + h) \cos \varphi \sin \lambda \quad (\text{Equation A-7-7})$$

$$z = \left(\frac{b^2}{a^2} N + h\right) \sin \varphi \quad (\text{Equation A-7-8})$$

where

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \quad (\text{Equation A-7-9})$$

## 2. From ECEF to LLA

There are two methods for this conversion

### 2.1. Iteration form

For  $\varphi$  and  $h$ , there is quick convergence for  $h \ll N$  starting at  $h_0 = 0$ .

$$\lambda = \arctan \frac{y}{x} \quad (\text{Equation A-7-10})$$

Starting with  $h_0 = 0$ .

$$\varphi_0 = \arctan \frac{z}{p(1 - e^2)} \quad (\text{Equation A-7-11})$$

While until  $h \ll N$ ,

$$N_i = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi_i}} \quad (\text{Equation A-7-12})$$

$$h_{i+1} = \frac{P}{\cos \varphi_i} - N_i \quad (\text{Equation A-7-13})$$

$$\varphi_{i+1} = \arctan \frac{z}{p \left( 1 - e^2 \frac{N_i}{N_i + h_{i+1}} \right)} \quad (\text{Equation A-7-14})$$

### 2.2. Closed form

$$\lambda = \arctan \frac{y}{x} \quad (\text{ Same as Equation A-7-10})$$

$$\varphi = \arctan \frac{z + e'^2 b \sin^3 \theta}{p - e^2 a \cos^3 \theta} \quad (\text{Equation A-7-15})$$

$$h = \frac{p}{\cos \varphi} - N \quad (\text{Equation A-7-16})$$

where

$$p = \sqrt{x^2 + y^2} \quad (\text{Equation A-7-17})$$

$$\theta = \arctan \frac{za}{pb} \quad (\text{Equation A-7-18})$$

## REFERENCES

- [1] Headquarters of Department of the Army, *FM 5-0 (previous FM 101-5) Army Planning and Orders Production*, Jan 2005. [Online], Available: <http://35.8.109.2/resources/FM5-0ArmyPlanningOrdersProd.pdf>, [Accessed: Jan, 2009]
- [2] David S. Alberts, John J. Garstka, Frederick P. Stein, *NETWORK CENTRIC WARFARE: developing and leveraging information superiority*, 2nd ed, CCRP, Feb 2000.
- [3] E. Shahbazian, *Introduction to DF: Models and Processes, Architectures, Techniques and Applications*, in *Multisensor Fusion*, Kluwer Academic Publishers, 2000, pp. 71-97.
- [4] R. R. Brooks, S. S. Iyengar, *Multi-sensor Fusion: Fundamentals and Applications with Software*, New Jersey; Prentice Hall, 1998.
- [5] S. Das, *High-Level Data Fusion*, Norwood, MA; ARTECH HOUSE, 2008.
- [6] M. E. Liggins II, C.Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, S. Thomopoulos, "Distributed Fusion Architectures and Algorithms for Target Tracking," in *Proceedings of the IEEE*, Vol. 85, No. 1, Jan. 1997.
- [7] D. McErlean, S. Narayanan, "Distributed Detection and Tracking in Sensor Networks," in *2002, Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems, and Computers*. Pacific Grove, CA., Nov. 2002.
- [8] T. G. Kim, C. Lee, E. R. Christensen, Bernard P. Zeigler, "System Entity Structuring and Model Base Management," *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 20, No. 5, Sep. 1990.
- [9] Bernard P. Zeigler, P. Hammonds, *Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange*, New York, NY; Academic Press, 2007.
- [10] Taekyu Kim, Younshin Han, Saehoon Cheon, Bernard P. Zeigler, "Network Traffic Analysis System Based on Data Engineering Methodology Using System Entity Structure," in *2007 International Conference on Convergence Information Technology*, Gyeongju, Republic of Korea, Nov. 2007.
- [11] Taekyu Kim, Chungman Seo, Bernard P. Zeigler, "Web Based Distributed SES/NZER Using Service Oriented Architecture," *SIMULATION: Transactions of The Society for Modeling and Simulation International*, submitted.
- [12] Saehoon Cheon, Doohwan Kim, Bernard P. Zeigler, "System Entity Structure For XML Meta Data Modeling; Application to the US Climate Normals," in *IEEE International Conference on Information Reuse and Integration*, Las Vegas, NV, July 2008.

- [13] Hojun Lee, Bernard P. Zeigler, "BML Enabled Information Exchange Framework in SES Ontology for C2", *GMU - AFCEA Symposium 2009*, Lansdowne, VA., May 2009.
- [14] Simulation Interoperability Standards Organization (SISO), *SISO-STD-008-200X, Coalition Battle Management Language (C-BML)*, [Online]. Available: <http://www.sisostds.org/index.php?tg=articles&idx=More&article=439&topics=102>. [Accessed: Jan. 10, 2009],
- [15] S. Carey, M. Kleiner, M. Hieb, R. Brown, "Standardizing Battle Management Language--facilitating coalition interoperability," paper 02E-SIW-005, in *Proceedings of the 2002 European Simulation Interoperability Workshop*, London, England, June 2002.
- [16] C. Turnitsa, "Battle Management Language: A Triangle with Five Sides," 06S-SIW-016, *2006 Spring Simulation Interoperability Workshop*, Huntsville, AL, Apr. 2006.
- [17] D. Kunde, A. Tolk, U. Schade, M. R. Hieb, "Harmonizing BML Approaches: Grammars and Data Models for a BML Standard," 09S-SIW-046, *2009 Spring Simulation Interoperability Workshop*, San Diego, CA, Mar. 2009.
- [18] U. Schade, M. R. Hieb, "Formalizing Battle Management Language: A Grammar for Specifying Orders," 06S-SIW-068, in *Spring Simulation Interoperability Workshop*, April 2006.
- [19] W. P. Sudnikovich, J. M. Pullen, M. S. Kleiner, S. A. Carey, "Extensible Battle Management Language as a Transformation Enabler," *SIMULATION: Transactions of The Society for Modeling and Simulation International*, Vol. 80, No. 12, pp. 669-680, 2004.
- [20] M. R. Hieb, U. Schade, "Formalizing Command Intent Through Development of a Command and Control Grammar." in *12th ICCRTS*. Newport, Rhode Island, June 2007.
- [21] K. Kruger, M. Frey and U. Schade, "Battle Management Language: Military Communication with Simulated Forces," RTO-MP-MSG-056, in *NATO RTO Modeling and Simulation Group Conference*, [Online]. Available: <ftp://ftp.rta.nato.int/PubFullText/RTO/MP/RTO-MP-MSG-056/MP-MSG-056-05.pdf>. [Accessed: Oct. 12, 2008]
- [22] U. Schade, M. R. Hieb, "Battle Management Language: A Grammar for Specifying Reports," *2007 Spring Simulation Interoperability Workshop*, Norfolk, VA., March 2007.
- [23] P. Hügelmeier, U. Schade, T. Zöller, "Application of BML to inter-agent communication in the ITSimBw simulation environment," in *Proceedings of the 2007 Winter Simulation Conference*, Washington, DC, December 2007.
- [24] U. Schade, J. Biermann, M. Frey, K. Kruger, "From Battle Management Language (BML) to Automatic Information Fusion," in *Proceedings on Information Fusion and Geographic Information Systems*, Springer, 2007.
- [25] MITRE Co., Cursor on Target, [Online]. Available: <http://cot.mitre.org/>. [Accessed: Mar. 2009]

- [26] M. Butler, *The Developer's Guide to Cursor on Target*, MITRE Corp., Aug. 2005.
- [27] D. Konstantopoulos, J. Johnston; "Data Schemas for Net-Centric Situation Awareness," in *2006 CCRTS*, San Diego, CA., June 2006.
- [28] A. Elledge, T. Zimmerman, B. Robinson; "Exchanging Link 16 CoT Messages and U.S. Army ABCS Messages via PASS," in *MILCOM 2007*, Orlando FL., Oct., 2007.
- [29] Merriam-Webster Online Dictionary, Ontology, [Online]. Available: <http://www.merriam-webster.com/dictionary/ontology>. [Accessed: Jan. 4, 2009].
- [30] M. M. Kokar, "Situation awareness: issues and challenges," in *Proceedings of the Seventh International Conference on Information Fusion*, Stockholm Sweden, July 2004
- [31] E. Blach, S. Plano, "Ontological Issues in Higher Levels of Information Fusion: User Refinement of the Fusion Process," *Fusion 2003: Proceedings of the sixth International Conference of Information Fusion*, Cairns, Queensland, Australia, July 2003.
- [32] World Wide Web Consortium (W3C), eXtensible Markup Language (XML), [Online]. Available: <http://www.w3.org/XML>. [Accessed Oct. 4, 2008]
- [33] Chungman Seo, Bernard P. Zeigler, "Interoperability between DEVS Simulators using Service Oriented Architecture and DEVS Namespace", in *Joint Symposium DEVS Integrative M&S (DEVS) and High Performance Computing (HPC)*, San Diego, CA., Mar. 2009.
- [34] World Wide Web Consortium (W3C), Web Service Architecture, [Online]. Available: <http://www.w3.org/TR/ws-arch/> [Accessed: Oct. 4, 2008]
- [35] World Wide Web Consortium (W3C), Web Services Description Language (WSDL), [Online]. Available: <http://www.w3.org/TR/wsdl/> [Accessed: Oct. 5, 2008]
- [36] Universal Description, Discovery and Integration (UDDI), [Online]. Available: <http://uddi.xml.org/> [Accessed: Oct. 5, 2008]
- [37] N. Baumgrtner, W. Retschitzegger, "Toward a Situation Awareness Framework Based on Primitive Relations," in *Proceedings on Information, Decision and Control 2007*, Adelaide, Australia, Feb. 2007
- [38] C. J. Matheus, K. Baclawski, M. M. Kokar, "Derivation of ontological methods in a situation awareness scenario," in *Proceedings of SPIE conference on Multi-sensor, Multi-source Information Fusion*, Orlando, FL, April 2003.
- [39] C. J. Matheus, "Using Ontology-based Rules for Situation Awareness and Information Fusion," in *W3C Workshop on Rule Languages for Interoperability*, Washington, D.C., Apr 2005.
- [40] A. N. Steinberg, "An Approach to Threat Assessment," in *2005 8<sup>th</sup> International Conference on Information Fusion*, Philadelphia, PA, July 2005.
- [41] M. M. Kokar, C. J. Maheus, K. Baclawski, "Ontology-based situatin awareness," *INFORMATION FUSION*, Vol. 10, Issue 1, pp 83-98, Jan. 2009.
- [42] World Wide Web Consortium (W3C), OWL Web Ontology Language, [Online], Aavailable: <http://www.w3.org/TR/owl-features/>, [Accessed: Feb. 2009].

- [43] The Rule Markup Initiative, RuleML, [Online] Available: <http://ruleml.org/> [Accessed: Mar 2009]
- [44] M. S. Grewal, A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, New York; John Wiley & Sons, 2001.
- [45] Y. Barshalom, X. Li, *Estimation and Tracking: principles, Techniques, and Software*, Norwood, MA, ARTECH HOUSE, 1993.
- [46] Y. Barshalom, X. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, New York, John Wiley & Sons, 2001.
- [47] Y. Barshalom, T. E. Formann, *Tracking and Data Association*, Orlando CA., Academic Press, 2000.
- [48] Y. Barshalom, X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, 1995.
- [49] X. Wang; Challa, S.; Evans, R., "Gating Techniques for Maneuvering Target Tracking in Clutter," *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 38, Issue 3, July 2002, pp.1087–1097.
- [50] MIP, *Overview of The C2 Information Exchange Data Model (C2IEDM)*, 6.1 ed, [Online]. Available: [http://www.e-mapsys.com/C2IEDM-MIP\\_Overview\\_20Nov2003.pdf](http://www.e-mapsys.com/C2IEDM-MIP_Overview_20Nov2003.pdf), Nov 2003.
- [51] Headquarters of Department of the Army; *FM 101-51/MCRP 5-2A (Operational Terms and Graphics)*, [Online] Available: [http://www.dtic.mil/doctrine/jel/service\\_pubs/101\\_5\\_1.pdf](http://www.dtic.mil/doctrine/jel/service_pubs/101_5_1.pdf), Sep. 1997. [Accessed: May 2009].
- [52] MIP, *The Joint C3 Information Exchange Data Model Overview*, 3.1b ed, [Online], Available: [http://www.mip-site.org/publicsite/04-Baseline\\_3.0/JC3IEDM-Joint\\_C3\\_Information\\_Exchange\\_Data\\_Model/JC3IEDM-Overview-GBR-DMWG-Edition\\_3.1b\\_2007-12-13.pdf](http://www.mip-site.org/publicsite/04-Baseline_3.0/JC3IEDM-Joint_C3_Information_Exchange_Data_Model/JC3IEDM-Overview-GBR-DMWG-Edition_3.1b_2007-12-13.pdf), Dec. 2007. [Accessed: May 2009].
- [53] RTSync, *SESBuilder*, Software, [Online], Available: <http://www.sesbuilder.com>.
- [54] World Wide Web Consortium (W3C), XML Schema, [Online]. Available: <http://www.w3.org/XML/Schema>. [Accessed: Nov. 21, 2008]
- [55] W3Schools, DTD, [Online], Available: [http://www.w3schools.com/DTD/dtd\\_intro.asp](http://www.w3schools.com/DTD/dtd_intro.asp). [Accessed: May. 22, 2009]
- [56] Arizona Center for Integrative Modeling and Simulation (ACIMS), *DEVJAVA*, Software, [Online], Available: <http://www.acims.arizona.edu/SOFTWARE/software.shtml>
- [57] E. Blasch, "Situation, Impact, and User Refinement," in *Proceedings of SPIE conference on Signal Processing, Sensor Fusion, and Target Recognition XII*, Orlando, FL., April 2003
- [58] E. Blach, "Level 5(User Refinement) issues supporting Information Fusion Management," in *the 9th International Conference on Information Fusion*, Firenze Italy, July 2006.

- [59] E. Blach, S. Plano, "Level 5: User refinement to aid the Fusion Process," in *Proceedings on SPIE on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2003*, Vol. 5099, April 2003.
- [60] W. T. Johnson, I. W. Dall, "From Knematics to Symbolics for Situation and Threat Assessment," in *Proceedings of 1999 Inforamtion, Decsion and Control*, Adelaide, Australia, Feb. 1999.
- [61] X. QingHua, L. Yingjie, L. FuXian, "Study on Knowledge Processing Techniques in Air Defense Operation Intelligent Aid Decsion," in *Proceedings of the Fifth International Conference on Computation Intelligence and Multimedia Applications*, Xi'an China, Sep, 2003.
- [62] World Wide Web Consortium (W3C), MathML, [Online] Available: <http://www.w3.org/Math/>, [Accessed: May 10, 2009]
- [63] Zeigler, B.P., Kim, T.G., and Praehofer, H., *Theory of Modeling and Simulation*, 2nd ed., New York, Academic Press, 2000.
- [64] Eclipse org., *eclipse*, Software, [Online], Available: <http://www.eclipse.org/>
- [65] The MathWorks, NIST; *JAMA: A Java Matrix Package*, Software, [Online], Availabe: <http://math.nist.gov/javanumerics/jama/>, [Acceesed: May 22, 2009]
- [66] BBN, *OpenMap*, Software, [Online], Available: <http://openmap.bbn.com/>, [Accessed: Mar. 2009]
- [67] u-blox, *Datum Transformations of GPS Positions: Application Note*, [Online], Available: <http://web.math.hr/~rogina/cartography.pdf> [Accessed: May 2009].