SISO-STD-011-2014

Standard for Coalition Battle Management Language (C-BML) Phase 1

Version 1.0

31 October 2013

Prepared by C-BML Product Development Group

P.O. Box 781238
Orlando, FL 32878-1238, USA

SISO Inc. Board of Directors
P.O. Box 781238
Orlando, FL 32878-1238, USA

Revision History

Version	Section	Date (MM/DD/YYYY)	Description
V1.0	All	10/31/2013	New

Participants

At the time this product was submitted to the Standards Activity Committee (SAC) for approval, the Coalition Battle Management Language (C-BML) Product Development Group had the following membership and was assigned the following SAC Technical Area Director:

Product Development Group

Saikou Diallo (Co-Chair)
Stan Levine (Co-Chair
Per Gustavsson (Vice-Chair)
Priscilla McAndrews (Secretary)

Jeff Abbott (SAC Technical Area Director)

Ababneh, Mohammad
Abbott, Jeff
Almen, Tobias
Asplund, Per
Barrett, Shelby
Beavin, William
Beeker, Emmet
Belz, Frank
Biagini, Marco
Birkel, Paul
Biwer, Mark
Bjorkman, Ulf
Blais, Curtis
Blood, Ben
Bonse, Remco
Briscoe, Derrick
Brook, Adam
Brown, Richard F.
Burgess, Pat
Can, Abdulkadir
Carlton, Bruce
Chartrand, Sidney
Chase, Tram
Choon, Bang
Clark, Nicholas
Collins, Darrell
Connor, Michael
Corner, Douglas
Crawford, Trevor
Croom-Johnson, Stella
Cross, Anthony
Danger, L.
De Champs, Patrick
de Reus, Nico
delaHostria, Em
Denehy, Greg
Devivi, Anthony
Dillman, Bradford
Dobbs, Verlynda
Dobrindt, Uwe
Dorman, Arren
Duvenhage, Arno
Elliott, D. Robert
Farmer, Gary
Fraka, Mike
Franceschini, Robert
Furuichi, Masakazu
Gagliano, Michael
Gagnon, Francois
Galvin, Kevin
Garcia, Juan
Garmendia-Doval, Beatriz
Gates, Buhrman
Gautreau, Bruno
Gauvin, Pierre
Giampapa, A.
Gonzalez Godoy, Sabas
Gougeat, Jean-Louis
Grayson, Stewart
Guillerit, Jean-Baptiste
Gupton, Kevin
Hansen, David
Hare, Carolyn
Hazen, Mark
Heath, Chris
Heffner, Kevin
Heilman, Eric
Hieb, Michael
Hill, Frank
Hollenbach, Jim
Hulten, Torbjorn
Hwang, Jae Jun
Ingraham, Lorie
Jones, Stephen
Jonsson, Fredrick
Kamath, Vikram
Kang, Jacqueline
Karlsson, Gunnar
Kelleher, Thomas
Lewis, Kenneth
Khimeche, Lionel
Kleiner, Martin
Knight, Sam
Krause, Lee
Krooks, David
Lacy, Lee
Le Leydour, Patrice
Liles, Stewart
Lind, Patrik
Lindegardh, Lars
Lindo, Wayne
Lofstrand, Bjorn
Lowe, Van
Lowe, Paul
Lue, Franklin
Magnusson, Klas
Mamaghani, Farid
Maroldo, Gina
Marrou, Lance
Martin, Edward
Martin, Kelvin
McCall, James
Mevassvik, Ole Martin
Morrison, John
Murray, William
Myjak, Michael
Nero, Eva
Neubauer, Peter
Niklas, Lisa
Niven, Mike
Nixon, Anthony
Nord, Henrik
Oates, William
Oliver, William

Peplow, Ken
Perez, Wilfredo
Perkins, Timothy
Perme, David
Peterson, Mark
Petty, Mikel
Porter, Ken
Powell, Edward
Prignac, Laurent
Pritchard, Ceri
Pullen, Mark
Radde, Guillaume
Rasch, Robert
Read, Robert
Reece, Douglas
Roberts, John
Rohlfing, Robert
Ross, Peter
Rouget, Chris
Salcedo, Claude
Sandberg, Stefan
Sayadi, Slim
Schade, Ulrich
Scrudder, Roy
Sexton, Dale
Shanks, Graham
Shellman, Ray
Shen, Edward
Shook, Tom
Singapogu, Samuel
Smith, Frederick
Sprinkle, Ronald
Stanzione, Thomas
Stenemo, Fannie
Stephens, Nelson
Stueve, Gerald
Sudnikovich, William
Tegner, Jan
Thiel, A.
Tolk, Andreas
Tudor, Grant
Tyson, Terry
Ullner, Fredrik
Verhage, Rene
Vernucci, Anthony
W. Albert
Whittington, Eric
Wittman, Robert
Wong, Kwok
Yang, David
Zalcman, Lucien

The Product Development Group would like to especially acknowledge those individuals that significantly contributed to the preparation of this product as follows:

PDG Drafting Group

Curtis Blais (Co-Editor)
Saikou Diallo (Co-Editor)

Ababneh, Mohammad
Brown, Richard F.
Chartrand, Sidney
Corner, Douglas
Heffner, Kevin
Levine, Stan
Priscilla McAndrews
Pullen, Mark
Scolaro, Dan
Singapogu, Samuel
St-Onge, Marc
Turnitsa, Chuck
Whittington, Eric

The following individuals comprised the ballot group for this product.

Ballot Group

Abbott, Jeff
Anglin, Dale
Blais, Curtis
Brook, Robert
Dobrindt, Uwe
Galvin, Kevin
Gautreau, Bruno
Gupton, Kevin
Gustavsson, Per
Jiménez López, Patricio
Marrou, Lance
Murray, William
Prignac, Laurent
Pullen, J. Mark
Sexton, Dale
Sheasby, Steven
Siegfried, Robert
Stone, George
Stoudenmire, Eugene
Stutzman, Marcy
Unrau, David
Whittington, Eric
Wittman, Robert

When the Standards Activity Committee approved this product on 18 February 2014, it had the following membership:

Standards Activity Committee

Jeff Abbott (Chair)
Marcy Stutzman (Vice Chair/Secretary)

Bailey, Grant
Blais, Curtis
Elaine Blount
Gravitz, Peggy
Gupton, Kevin
Lowe, Paul
McGlynn, Lana
McLean, Angus L M Thom, III
Oates, William
Riggs, Bill
Youngblood, Simone

When the Executive Committee approved this product on 14 April 2014, it had the following membership:

Executive Committee

Michael O'Connor (Chair)
James Coolahan (Vice Chair)
Jane Bachman (Secretary)

Abbott, Jeff
Bouwens, Chris
Daly, John
Graham, David
Gustavson, Paul
Igarza, Jean-Louis
Morse, Katherine
Scrudder, Roy
Whittington, Eric

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

The Coalition Battle Management Language (C-BML) is a standard language for expressing and exchanging plans, orders, requests, and reports across command and control (C2) systems, live, virtual and constructive (LVC) modeling and simulation (M&S) systems, and autonomous systems participating in Coalition operations.

There is a long history of military requirements, research projects, technical reports, journal articles, and conference papers identifying the need for improving interoperation of Command and Control (C2) and Modeling and Simulation (M&S) systems [5]. The development of digitized C2 systems and the use of M&S to support "train-as-you-fight" as well as Course of Action Analysis (COAA) and Mission Rehearsal have created an increased requirement for interoperability across these systems. In addition, the emergence of autonomous systems and the move to net-centric and network-enabled operations create new opportunities and context within which M&S can support the warfighter and civilian command posts. Adding to the interoperability challenge, major military and civilian operations are no longer conducted by single services, agencies, or organizations in a single country. Rather, they are increasingly multi-national, multi-service, and multi-agency and likely to be conducted within a coalition or or collaboration of organizations. These complexities drive the requirements for multi-national interoperability and the development of standards for inter-system information exchange.

This C-BML standard identifies the underlying data model and provides specification of an information exchange content and structure for C-BML expressions. The data model is based on the Joint Consultation Command and Control Information Exchange Data Model (JC3IEDM) [8-22].

Overall, the C-BML standard is being developed in three phases; namely:

- Phase 1, Data Model: Phase 1 of the C-BML standard describes a sufficient data model to unambiguously define a set of military orders using JC3IEDM as a starting point and extending it as necessary so that the orders can be interpreted by C2, M&S, and ultimately autonomous systems. This C-BML Specification describes the data model as a subset of JC3IEDM and specifies the information exchange content and structure in the form of an Extensible Markup Language (XML) schema.
- Phase 2, Formal Structure (Grammar): Phase 2 will introduce a grammar (syntax, semantics, and vocabulary) as part of the information exchange content and structure specification. The objective is to formalize the definition of tasks such that they are rigorous, well documented, and parseable. Refer to [B30, B31] for examples of work in this area. The need for a grammar for tasking and reporting is seen as a common requirement for both the C-BML and Military Scenario Definition Language (MSDL) [6] standardization efforts.
- Phase 3, Formal Semantics (Ontology): Phase 3 will include development of a battle management ontology to enable conceptual interoperability across systems (see [B11] for some early work in this area).

The Phase 1 standardization effort also provides an accompanying document, the C-BML Guidelines (in preparation), providing examples of the use of the Phase 1 specification to facilitate work by early adopters of the standard.

A closely related standard in SISO is the Military Scenario Definition Language (MSDL, SISO-STD-007-2008) [6]. MSDL provides a common representation of scenario information that can be exchanged across multiple C2 and M&S systems [B8]. Intended for use in initializing various systems, MSDL describes the physical location and setting of a scenario (e.g., terrain, weather), forces and force structures, control measures, and other information. MSDL is used to identify forces and scenario settings that can be used in C-BML expressions of plans, orders, requests, and reports. Furthermore, scenario descriptions can include initial sets of plans and orders (e.g., air tasking orders, ground movement orders, and ship-to-shore landing plans), which will be expressed using the C-BML standard. Numerous papers have been presented at the SISO Simulation Interoperability Workshops (SIWs) relating to the close interrelationship between C-BML and MSDL [B1, B2, B25, B26, B27].

1.1 Purpose

Fundamentally, when two systems need to exchange information, one system sends the information to the other through some communications medium, as depicted below.

Figure 1: Generic system-to-system interaction

Several configurations are possible. System A could be a C2 system passing an order to a simulation system (System B) to be executed in the simulation environment. System A could be a constructive simulation system passing synthetic target data to a virtual simulation (System B). System A could be an autonomous system providing situation report data to a C2 system (System B). Many other such combinations apply, but they all share the same fundamental notion. Currently, there are many formats for the information transferred between and among such systems. Some of the formats are standardized and used by many systems; some are specialized and used by a small number of systems. In the worst case, two specific systems interact using a unique point-to-point information format only applicable to that pair of systems.

In the case of transfer of plans, orders, requests, and reports across C2, M&S, and autonomous systems, the C-BML concept, very simply, is standardization of the structure, content, and mechanism for this information exchange, as shown in Figure 2.

Figure 2: System-to-system interaction using C-BML

In military operations, the nature (format and content) of the information to be exchanged is determined by the doctrine governing the exchange. Certain forces conducting certain operations are required by doctrine to provide certain information. The “C-BML expression” in the diagram above essentially encapsulates the doctrinal exchange. Said another way, for a system to employ C-BML, its doctrinal expressions, in whatever form (e.g., military message formats) would be transformed into C-BML expressions, either internally within the sending system or through some adaptor external to the sending system. If all systems adopt the C-BML standard, then only C-BML expressions will be transmitted between and among systems when transferring plans, orders, requests, and reports.

The purpose of C-BML standardization is to specify requirements for the generation and transfer of C-BML expressions. *Generation* of C-BML expressions depends upon two parts of the standard: (1) the C-BML data model and (2) the information exchange structure and content specification. Together these describe the lexical building blocks for construction of C-BML expressions. *Transfer* of the C-BML expressions across systems employs the third part of the C-BML standard; namely, the information exchange mechanism (to be specified in a future effort). This concept for employment of the C-BML standard is summarized in Figure 3.

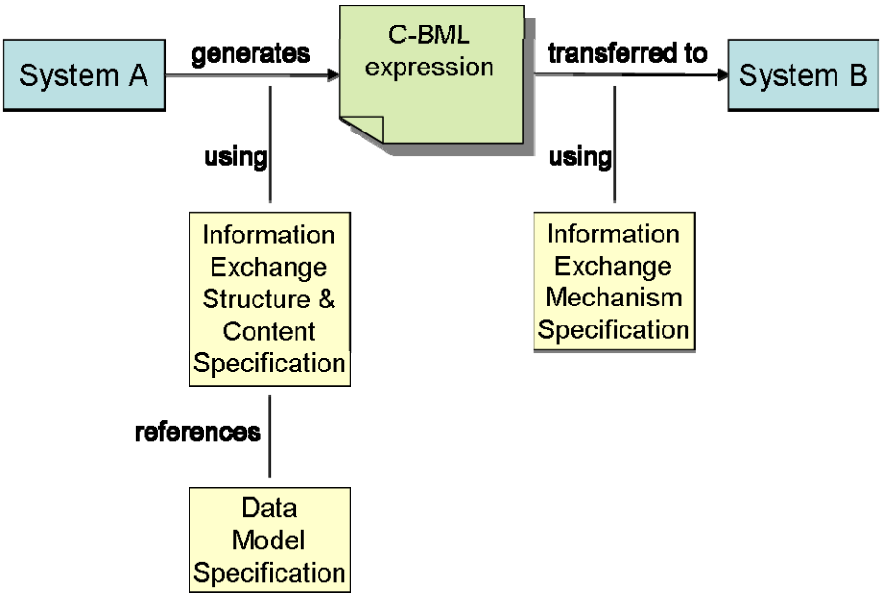

Figure 3: Concept of employment of the C-BML standard

1.2 Scope

This specification (Phase 1 C-BML) defines the C-BML in terms of (1) a standard data model and procedures for extending the data model and (2) a description of the information components of the language. This specification does not attempt to provide the structure and content for all C-BML expressions that may be needed by the broad user community; but provides an initial set of constructs that can be used by the community to create initial implementations of the standard as a way of informing further development and enrichment of the standard over time. Follow-on phases in the C-BML standard development process will augment and extend this initial specification with (1) description of a formal grammar for plans, orders, requests, and reports (Phase 2) and (2) specification of formal semantics of the language (Phase 3). A separate Phase 1 C-BML Guidelines document (in preparation) provides examples of application of this specification to aid in early adoption of the standard across C2, M&S, and autonomous system communities.

1.3 Objectives

The primary objective of this specification is to provide sufficient information to enable early adopters of the Phase 1 C-BML standard to construct and exchange C-BML orders, requests, and reports.

1.4 Intended Audience

The C-BML Specification is intended for use by software developers (specification, design, implementation, and test) in the C2, M&S, and autonomous system domains. The document is a means for familiarizing developers with the fundamental concepts and building blocks of the language. The associated Phase 1 C-BML Guidelines document and supporting files provide discussion and examples of ways to employ the Phase 1 specification.

2 References

2.1 SISO Documents

	Document Number	Title
1	SISO-ADM-001-2011	Policy for Numbering of SISO Products
2	SISO-ADM-002-2011	SISO Policies and Procedures
3	SISO-ADM-003-2011	SISO Balloted Products Development and Support Process
4	SISO-ADM-005-2011	Policy for: The Style and Format of SISO Documents
5	SISO-REF-016-2006	Coalition Battle Management Study Group Final Report
6	SISO-STD-007-2008	Standard for: Military Scenario Definition Language (MSDL)

2.2 Other Documents

	Document Number	Title
7	IEEE 1320.2-1998	IEEE 1320.2-1998. IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X
8	JC3IEDM Overview – DMWG 20090514 Edition 3.0.2	Overview of the Joint C3 Information Exchange Data Model (JC3IEDM Overview)
9	JC3IEDM MAIN – DMWG 20090514 Edition 3.0.2	The Joint C3 Information Exchange Data Model (JC3IEDM Main)
10	JC3IEDM – Annex A – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex A Glossary
11	JC3IEDM – Annex B – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex B Entities
12	JC3IEDM – Annex C – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex C Attributes
13	JC3IEDM – Annex D – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex D Relationships
14	JC3IEDM – Annex E – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex E Domain Values
15	JC3IEDM – Annex F – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex F Other Domains
16	JC3IEDM – Annex G1-Text – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex G1 Compendium of Text Business Rules
17	JC3IEDM – Annex H – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex H Naming Conventions and Class Words
18	JC3IEDM – Annex I – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex I Summary of IDEF1X Data Modelling Methodology and Notation
19	JC3IEDM – Annex K – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex K Logical View
20	JC3IEDM – Annex O – DMWG 20090514 Edition 3.0.2	JC3IEDM Annex O Extensible Markup Language (XML) Reference Schemas and Implementation Guidance
21	JC3IEDM – Metamodel – DMWG 20090514 Edition 3.0.2	The Joint C3 Information Exchange Data Model Metamodel (JC3IEDM Metamodel)
22	JC3IEDM-GuideToCP-DMWG 20090514 Edition 3.0.2	Guide to Change Proposals for MIP Data Specifications (JC3IEDM-GuideToCP)
23	http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/	XML Schema Part 1: Structures Second Edition World Wide Web Consortium (W3C) W3C Recommendation 28 October 2004

3 Definitions

C-BML Compliance: Systems shall generate, exchange, and process valid C-BML expressions, where validity is based on fulfillment of the structure and content requirements as specified in the C-BML XML schemas. Source: This standard.

C-BML Data Model: Identification of concepts and attributes describing those concepts used to construct valid C-BML expressions. Source: This standard.

C-BML Expression: An order, request, or report stated using the C-BML information exchange content and structure. Source: This standard.

C-BML Grammar: A lexicon and set of rules for generating valid C-BML expressions. (Refer to the definition of "Formal Grammar" below.) Source: This standard.

C-BML Information Exchange Content: Identification of the information elements constituting C-BML expressions. Source: This standard.

C-BML Information Exchange Mechanism: Description of a technical means for constructing, exchanging, and parsing C-BML expressions. Source: This standard.

C-BML Information Exchange Structure: Description of the syntax of valid C-BML expressions and information components of C-BML expressions. Source: This standard.

C-BML What: C-BML information component identifying action(s) to be performed or effects to be delivered (order or request) or action(s) and effect(s) that have occurred (report). Source: This standard.

C-BML When: C-BML information component describing the timeframe in which action(s) or effect(s) is to occur (order or request) or when action(s), effect(s), or event(s) has occurred (report). Source: This standard.

C-BML Where: C-BML information component providing the location of an object in the battlespace (*C-BML Who*), the location where action(s) or effect(s) is to occur (order or request), or the location where action(s), effect(s), or event(s) has occurred (report). The location may be a complex object, such as an area or a sequence of locations. Source: This standard.

C-BML Who: C-BML information component identifying the battlespace object (1) that is directed to perform action(s) to deliver effects (as defined in an order), (2) that has been requested to perform action(s) to deliver effects (as identified in a request), (3) that has been observed or has performed actions or delivered effects (as described in a report), or (4) on which actions or effects are to be performed (e.g., target). Source: This standard.

C-BML Why: C-BML information component describing the rationale or purpose of action(s) to be performed or effect(s) to be delivered (order or request), or the desired end state of such action(s) or effect(s). Source: This standard.

Coalition: A collection of actors from diverse nations, military services, and organizations that is operating cooperatively and in coordination to achieve a shared objective; also a collection of their C2 and simulation systems that interoperate using C-BML. Source: This standard.

Coalition Battle Management Language: a standard language for expressing and exchanging orders, reports, and requests across command and control (C2) systems, live, virtual and constructive (LVC) modeling and simulation (M&S) systems, and autonomous systems participating in Coalition operations. Source: This standard.

Commander's Intent: Intent is the state of mind with which an act is done. Source: [B17]

Intent in its broadest interpretation can be seen as the purpose and all its implications. Intent is the mechanism to explain how to fulfill the mission expressed in an order (also an eligible plan or request). Intent can be explicitly or implicitly communicated. Source: [B10]

The intent defines the end-state in relation to the factors of mission: adversary, operating environment, terrain, forces, time and preparation for future operations. As such, it addresses what results are

expected from the operation, how these results might enable transition to future operations, and how, in broad terms, the Commander expects the force to achieve those results. Its focus is on the force as a whole. Additional information on how the force will achieve the desired results is provided only to clarify the Commander's intentions. Source: [B22]

The commander's intent is a clear, concise statement of what the force must do and the conditions the force must establish with respect to the enemy, terrain, and civil considerations that represent the desired end state. Source: [B7]

Composite: Definable, delineable, and conceptually related components of the language. Source: This standard.

Conceptual Data Model: Description of data using concepts and terminology from the application domain, such as military unit table of equipment and information content of military orders. Source: This standard.

Conceptual Schema: A schema of the American National Standards Institute Standards Planning and Requirements Committee (ANSI/SPARC) Three Schema Architecture, in which the structure of data is represented in a form independent of any physical storage or external presentation format. Source: [B19]

Conceptual Interoperability: The ability of multiple systems to exchange information, with the assurance of common and consistent interpretation and application of that information, not through separately developed implementations of processing logic, but through shared formal semantic descriptions of the information being exchanged. Source: This standard.

Doctrine View: Expressions of plans, orders, requests, and reports using terminology particular to a specific nation, service, or organization. Source: This standard.

Formal Grammar: A formal grammar (sometimes simply called a grammar) is a set of formation rules for strings in a formal language. The rules describe how to form strings from the language's alphabet that are valid according to the language's syntax. A grammar does not describe the meaning of the strings or what can be done with them in whatever context, only their form. Source: [B28]

Interoperability: The ability of two or more systems or components to exchange information and use the information that has been exchanged. Source: [B14]

Lexical Functional Grammar: Lexical Functional Grammar (LFG) was first mentioned in [B3] and is a context free grammar. The three words of LFG characterize the language building structure: *Lexical* means that structural information is stored in the lexicon, in particular, verb frames; *Functional* refers to subject and objects and how they are grouped together by attribute-value matrices (i.e., how the lexical elements are to be used in context); and *Grammar* is how the structure of the language is built. The syntax is basically defined in a constituent structure (C-structures), functional structure (F-structures), and semantic roles (A-structures). The motivation for LFG is that many phenomena are thought to be more naturally analyzed in terms of grammatical functions, rather than on the level of constituent structure (i.e., a word or group of words that is treated as a single unit within a hierarchical structure and organized within a syntactic tree). Source: [B3, B4, B15, B28, B29]

Logical Data Model: A model that provides a common dictionary of data definitions to consistently express models wherever logical-level data elements are included in the descriptions. Source: [B18]. Also, a description of information using constructs such as classes and properties, without committing to a particular implementation. Source: This standard.

Military Scenario: A specific description of the situation and course of action at a moment in time for each element in the scenario. The description of the scenario conveys reality (what is true about the situation, such as the forces identified as participants in the situation) and perceived reality (what is considered to be true based on intelligence information). Source: [6].

Mission: (1) A clear, concise statement of the task of the command and its purpose. (2) One or more aircraft ordered to accomplish one particular task. Source: [B23]. Also, a sequence of tasks to be executed in a coordinated manner. Source: This standard.

Ontology: A formal specification of a conceptualization. Source: [B9]

Order: A singular directive from a competent authority designating an action to be performed by a specified actor. Note: An Operations Plan (OPLAN) differs from an Operations Order (OPORD) in that the OPLAN states critical assumptions that form the basis of the plan and time of execution is not introduced. It is imperative that these assumptions be revalidated to describe the operational situation needed to transform the OPLAN into an executable OPORD. The OPLAN becomes an OPORD when the conditions of execution occur and an execution time is determined.The essentials of an OPLAN/OPORD are: situation, mission, commander's intent, concept of operations, schema of maneuver and main effort, the assignment of tasks/missions to formations/units, the support and assistance to be provided, and command and signal instructions. An OPORD should include only such detail as is necessary for commanders of subordinate formations/units to issue their own orders and to ensure coordination. The detail of how supporting and specialist units are to carry out their tasks should be issued in their own orders, and will use the same format as an OPORD unless otherwise specified. Source: [B21]

Physical Data Model: A model that defines the structure of the various kinds of system or service data that are utilized by the systems or services in the architecture. Source: [B18]. Also, a description of information in an implementable, executable representation, such as a specific data schema defining tables and attributes implemented in a specific relational database management system. Source: This standard.

Plan: A collection of orders designed to be performed in cooperative and coordinated fashion to mutually contribute to achievement of a desired objective. A plan can also be considered as a proposal for executing a command decision or project. It represents the command's preparation for future or anticipated operations. Because plans concern future operations and help the staff make assumptions about the nature of the situation at the time of execution, they cannot remain static. As the commander and staff change or adjust their estimates to reflect the current analysis of this situation, they must also change the plans to reflect the results of this analysis. Source: [B21]

Protocol View: Rules for exchanging plans, orders, requests, and reports that are agreed upon by two or more systems. Generally, standardized protocols are preferred to facilitate participation by many systems. Source: This standard.

Report: An account or statement describing in detail an event, situation, or the like, usually as the result of observation, inquiry, etc. Source: Dictionary.com. Also, information about some actor, action, or event generated by a system to inform human users, other components of the originating system, or other systems. Source: This standard.

Representation View: Selection of a conceptual, logical, or physical data model to use in expressing plans, orders, requests, and reports. If the representation view (data model) uses different terminology than the doctrine view, some lossless transformation is required between the two views to convert doctrinal expressions to data expressions and vice versa. Source: This standard.

Request: The act of asking for something to be given or done. Source: Dictionary.com. In JC3IEDM, Request is an ACTION-TASK that states a requirement. Source: [11]. Also, an appeal for an action from one organization to another, where the former has no direct authority over the capabilities or resources of the latter. Source: This standard.

Schema: Definition of data structure. Source: [B19]

Task: A clearly defined piece of work, sometimes of short or limited duration, assigned to or expected of a person. Source: Dictionary.com. Also, a specific action to be performed, possibly at a specified location and time by a specified actor. A task may be defined solely as an action, but before execution must be assigned an actor, location, and/or time. Source: This standard.

4 Acronyms and Abbreviations

AAP	Allied Administrative Publication
ACO	Airspace Control Order
ANSI/SPARC	American National Standards Institute Standards Planning and Requirements Committee
ATO	Air Tasking Order
BC	Battle Command
BML	Battle Management Language
C2	Command and Control
C2IEDM	Command and Control Information Exchange Data Model
C-BML	Coalition Battle Management Language
CCSIL	Command and Control Simulation Interface Language
COAA	Course of Action Analysis
CRM	Common Reference Model
CSL	Condensed Scripting Language
DBMS	Database Management System
DG	Drafting Group
EXI	Efficient XML Interchange
GH	Generic Hub
HLA	High Level Architecture
HTTP	Hypertext Transfer Protocol
IDEF1X	Integration Definition for Information Modeling
JC3IEDM	Joint Consultation, Command, and Control Information Exchange Data Model
LVC	Live, Virtual, Constructive
M&S	Modeling & Simulation
MIP	Multilateral Interoperability Programme
MSDL	Military Scenario Definition Language
MSG	Modeling and Simulation Group
NATO	North Atlantic Treaty Organization
NIST	National Institute of Science and Technology
OID	Object Identifier
OPLAN	Operations Plan
OPORD	Operations Order
PDG	Product Development Group
PSG	Product Support Group
SAC	Standards Activity Committee
SFA	Sub-Functional Area
SG	Study Group
SISO	Simulation Interoperability Standards Organization
SIW	Simulation Interoperability Workshop
SOAP	Simple Object Access Protocol
STANAG	Standardisation Agreement
STD	Standard
STOW	Synthetic Theater of War
UAV	Unmanned Aerial Vehicle
UDDI	Universal Description, Discovery, and Integration
URL	Universal Resource Locator
URN	Universal Resource Name
5Ws	Who, What, When, Where, Why
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
XBML	Extensible Battle Management Language
XML	Extensible Markup Language
XSBC	XML Schema-based Binary Compression
XSD	XML Schema Document

5 Coalition Battle Management Language Concept

C-BML provides an unambiguous method for conveying orders and commands to live, simulated, and autonomous forces. C-BML formalizes concepts such as the "Who, What, When, Where, Why" (5Ws) needed to command and control forces. These constructs must be understood by C2 systems, simulations, and autonomous systems.

These principles have led researchers to describe three "views" or perspectives on the nature of a battle management language [B33]:

- Doctrine View: Every term within the language must be unambiguously defined and must be rooted in military doctrine. The C-BML standard should not reflect a single service doctrine, but reflect the common doctrinal components that are needed for the coalition participants to effectively execute operational and training missions. This is conveyed in C-BML by generic information elements from which doctrine-specific expressions can be constructed. Groundbreaking work performed for the US Army was documented in [B5] and [B32]. Previous work performed provided recommendations to the development of the C-BML standard and are sources of methods and procedures that can be followed by future C-BML developers.
- Representation View: The representation view structures and relates the terms defined in the doctrine in a way that can result in the description of information contained in missions, tasks, requests, and reports. Relevant representations can include conceptual, logical or physical data models or fully formalized ontologies. Many prototypes and experiments have used the JC3IEDM as the underlying data model for the language [B16].
- Protocols View: Protocols standardize the rules by which information is transported from the source system (C2, simulation, or autonomous) to the target system (C2, simulation, or autonomous). In the emerging net-centric operational environment, Web-based standards and grid standards offer some candidate protocols. In particular, the use of XML to describe information exchange requirements is considered fundamental since it is the currently accepted standard for data description across battle command (BC), simulation, and autonomous systems. The Extensible Battle Management Language (XBML) project (and follow-on efforts) used Hypertext Transfer Protocol (HTTP)-based Web services as the means for communications across distributed applications [B12, B13]. Based on results from other programs, as well as other interested experts in the domain of application of Web services within computer grids [B24], solutions that are more general may be needed in the international domain, which further point to XML. Many have expressed concern that the size of XML files will over-burden already limited bandwidth supporting military operations. These concerns are being addressed through activities such as the World Wide Web Consortium (W3C) Efficient XML Interchange (EXI) Working Group (see http://www.w3.org/XML/EXI) and numerous commercial products that have demonstrated the ability to further reduce the size of transmitted XML files compared to standard text compression techniques [B20].

Figure 4 [B33] summarizes the three views described above. This C-BML Specification focuses on aspects of the representation and protocol views through identification of the data model and initial specification (within scope of Phase 1 development efforts) of the information exchange structure and content.

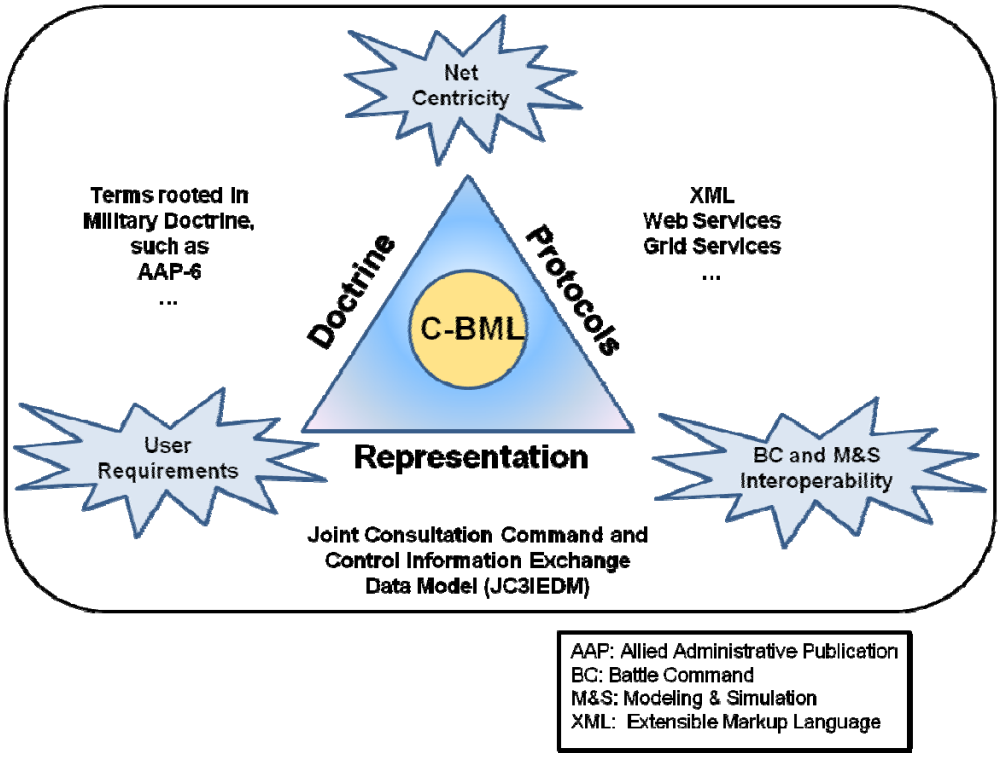

Figure 4: Three sides of the C-BML triangle: doctrine, representation, and protocol

Based on the foundation built on previous BML conceptual efforts and research, this Phase 1 C-BML Specification seeks to provide a starting point for practical adoption of the principles of a battle management language in the context of Coalition (multi-national, multi-organizational) system interoperability. The next section describes the initial data model (representation view) and initial information exchange content and structure (an abstraction of the doctrine view) to begin formalizing the C-BML standard.

6 C-BML Data Model

This section describes the data model selected for the Phase 1 C-BML specification; namely, the JC3IEDM Baseline Edition 3.0.2 [8-9]. This data model serves as a common reference model (CRM) to provide the underlying representation view of C-BML defining the basic lexicon for construction of C-BML expressions. In the early stages of C-BML, it was recognized that the JC3IEDM was a well-established international data model that provides many data constructs that are useful in the expression of plans, orders, reports, and requests. While JC3IEDM can provide a starting vocabulary for C-BML that has been vetted by the MIP-member countries, it does not define the macro-level constructs needed for expressing plans, orders, reports, and requests for broad application to C2 system and M&S system interchange that will be specified in the C-BML standard as it evolves through its three-phase development process. JC3IEDM is an evolving standard in its own right; governance of the C-BML standard may involve periodic assessment of changes in JC3IEDM to determine if they are beneficial to the evolving C-BML standard. Changes to the SISO C-BML standard, whether stimulated by evolution of JC3IEDM or evolving needs of the C-BML user community, will be managed by the future C-BML Product Support Group (PSG) and any continuing C-BML PDG/DG organizations. This section includes discussion of the need for processes to address extensions to JC3IEDM identified by users of C-BML. As both standards evolve, coherence will be maintained through respective change control processes; i.e., the SISO change process through established PDG and PSG activities [2, 3] and the MIP change process laid out in the JC3IEDM Guide to Change Proposals [22].

6.1 Base Model: JC3IEDM

JC3IEDM is the central reference model for initial specification of C-BML. JC3IEDM is sufficiently robust to handle much of the required data in command and control data exchanges across the systems that C-BML is intended to serve (C2, M&S, and autonomous systems). Within the international M&S and C2 communities, there is widespread acceptance of the JC3IEDM as the standard basis for interchanging command and control information. JC3IEDM serves as a neutral, independent data model that belongs to no single system or community (branch of service or nation of origin), but can serve to describe the information exchanged between systems that C-BML is interested in (again, C2, M&S, and autonomous systems).

6.1.1 Model Overview

The JC3IEDM data model comprises two categories of information: the Generic Hub (GH) and the Sub-Functional Areas (SFAs). The data model encompasses information from multiple functional areas in the domain of military operations. All common data or, better said, all data that need to be exchanged by at least two functional areas, become part of the Generic Hub. The remaining data are modeled as extensions of the Generic Hub data into the Sub-Functional Areas.

Initial evolution of the JC3IEDM under the MIP included specific inputs from the following functional areas: conventional fire support, barrier engineering operations, communications and electronics, and personnel administration. Operational requirements have been drawn from these as well as other areas, as documented by the MIP (http://www.mip-site.org). JC3IEDM describes objects of interest on the battlefield; e.g., organizations, persons, equipment, facilities, geographic features, weather phenomena, and military control measures such as boundaries, using a common and extensible data modeling approach.

Fundamental information in JC3IEDM includes OBJECT-TYPE, OBJECT-ITEM, ACTION, and CAPABILITY [9]. For example, the battlefield consists of a large number of objects, each with its own set of characteristics. Objects may be described as a class or type rather than as individually identified items: types or kinds of objects are recorded as OBJECT-TYPE, while specific instances of an object type are identified by use of OBJECT-ITEM. General attributes are collected on the type side, such as capabilities; only the instantiation-specific values are given on the item side. For example, the characteristics of the weapon are specified on the type side, while the actual ammunition state and location of a specific weapon are provided on the item side. The top-level relationship between OBJECT-TYPE and OBJECT-

ITEM is shown in Figure 5.[1] The figure also shows the major subcategories of OBJECT-TYPE and OBJECT-ITEM represented in the data model. The subcategories are given by a category code attribute. In addition to the values of FACILITY, FEATURE, MATERIEL, ORGANISATION, and PERSON, there is also an entry of "Not known" for an OBJECT-ITEM which is tracked but has not yet been classified. If additional categories are needed to meet a particular application requirement, the model can be extended through a process managed by the MIP. Extensions may involve adding new attribute values (e.g., new category and sub-category codes for OBJECT-TYPE and OBJECT-ITEM), adding new attributes, adding new tables, or adding new associations.

The following paragraphs provide some additional information as a brief introduction to JC3IEDM structure and terminology. It is beyond the scope of this specification to provide a complete description of the JC3IEDM. The reader is asked to refer to the full set of JC3IEDM documentation [8-22]. The JC3IEDM documentation is a normative reference for the C-BML standard.

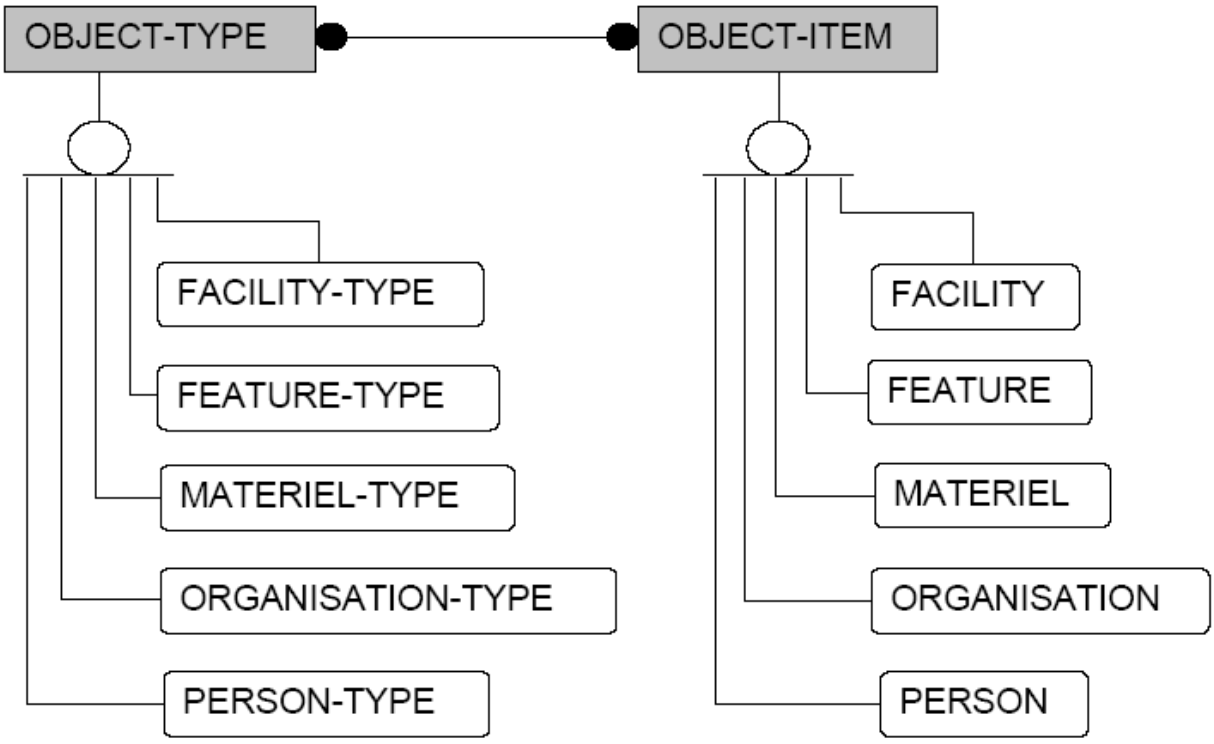

Figure 5: JC3IEDM Object-Type and Object-Item entities

6.1.2 Model Aspects

Principal components of the data model are entities (commonly implemented as tables in a database management system implementation of the model) [11], attributes (fields in the tables) [12], relations (relational links between the different tables) [13], and the rules of intended use [16]. These aspects of JC3IEDM are briefly described below.

6.1.2.1 Entities

An entity represents a discrete object in the structure of the data model. Entities can be thought of as nouns. Within the JC3IEDM, entities can, among other things, be representative of some physical thing (OBJECT-ITEM), a class of items (OBJECT-TYPE), some process (ACTION-TASK), or the results of

[1] The C-BML specification retains the British English spelling of terms used in the JC3IEDM, such as "ORGANISATION."

some process (ACTION-EFFECT). The JC3IEDM also uses some relationship entities to grant attribution to a relationship (known within the JC3IEDM as an *association*) between two (or more) other entities; relations are described below in paragraph 6.1.2.3.

6.1.2.2 Attributes

Entities and relationships can both have attributes. Attributes are the data elements associated with either the entity or the relationship. Some of these are required for the identity of the entity or relationship. Some examples include physical characteristics for OBJECT-TYPE entities or index values for relationships. Within a database implementation of the model, attributes are the fields within a table.

6.1.2.3 Relations

Some entities have semantic links to other entities, in order to represent a more complex idea than can be represented in a single entity. Within the JC3IEDM, some of these relations are required and some are optional. In all cases, when a relation exists, the entities so joined become a relationship. All relationships in the JC3IEDM have a phrase that defines the relationship; some examples of these are “is-specified-as” and “is-the-object-of.” As introduced earlier, these relationships are referred to as *associations* within the data model.

If the relationship requires attribution, then there may be an intermediary entity that exists only to hold that attribute. For convenience, these may be referred to as relationship-entities. Some examples of a relationship-entity are ACTION-TASK-RULE-OF-ENGAGEMENT (combining ACTION-TASK and RULE-OF-ENGAGEMENT) and CANDIDATE-TARGET-DETAIL-ASSOCIATION (which combines two CANDIDATE-TARGET-DETAIL entities, one as the subject of the other).

When the data model is implemented in a relational database, the foreign key mechanism is used to designate associations (relations). If the relationship requires attribution, then a relation between the associated entities and the relationship-entity (which contains the attribute) is made, and all are associated via the use of indexes as attributes.

6.1.2.4 Rules of Intended Use

The JC3IEDM documentation available from the MIP provides guidance and rules about how the above elements are used and when they are and are not required to be present [8, 9, 16]. Examples in the documentation cover many of the common uses for the model (the most common forms of command and control data exchange – including tasks, reports, and others), and how the correct elements must be employed. Sequence and structure, where important, are noted, as well as the use of mandatory fields.

7 C-BML Information Exchange Content and Structure

7.1 Conceptual Model of the Language

The principal information components of C-BML are the 5Ws: Who, What, When, Where, and Why. In the abstract, this information is fundamental to the expression of orders, requests, and reports for any doctrine of any service or agency, of any nation. The following constitute a definition of the 5Ws for purposes of this Phase 1 C-BML Specification:

- *Who:* C-BML information component identifying the battlespace object (1) that is directed to perform an action (as defined in an order), (2) that has been requested to perform an action (as identified in a request), (3) that has been observed or has performed an action (as described in a report), or (4) on which an action is to be performed (e.g., target).

- *What:* C-BML information component identifying action(s) to be performed or effects to be delivered (order or request) or action(s) and effect(s) that have occurred (report).

- *When:* C-BML information component describing the timeframe in which action(s) or effect(s) is to occur (order or request) or when action(s), effect(s), or event(s) has occurred (report).

- *Where:* C-BML information component providing the location of an object in the battlespace (*C-BML Who*), the location where action(s) or effect(s) is to occur (order or request), or the location where action(s), effect(s) or event(s) has occurred (report). The location may be a complex object, such as an area or a sequence of locations.

- *Why:* C-BML information component describing the rationale or purpose of action(s) to be performed or effect(s) to be delivered (order or request), or the desired end state of such action(s) or effect(s).

The 5Ws relate to the C-BML doctrine view (introduced in Section 5). This abstraction of fundamental elements in the content of doctrinal expressions of orders, requests, and reports facilitates employment by any service, organization, or nation.

These fundamental components of the language are used to construct C-BML expressions. A *C-BML expression* is an order, request, or report constructed in accordance with the C-BML information exchange content and structure specification OR constructed in accordance with a content and structure specification that uses constructs from the C-BML information exchange content and structure specification. As stated in the Scope section, this Phase 1 C-BML Specification does not attempt to define the structure of all possible C-BML expressions that may be needed across the C2, M&S, and autonomous system communities; but attempts to provide an approach and building blocks for construction of a wide range of expressions needed across those communities. To facilitate early adoption, this specification does provide an initial structure for a certain class of expressions that have proven useful in prototype implementations over the past decade.

In the expression of orders, requests, and reports, each “W” has a usage particular to that expression. For example, in an order, a “Who” may identify the authority giving the order, while another “Who” identifies the organization that will carry out the order. Examples of the usage of the 5W’s in orders, requests, and reports are provided in Table 1.

Table 1: Example usage of the 5W's in orders, requests, and reports

W	Used In	Name	Definition
Who			
	Orders		
		TaskeeWho	Specifies who is executing the task
		TaskerWho	Specifies who is ordering or authorizing execution of the task
		AffectedWho	Specifices a "who" affected by the task to be performed
	Reports		
		ReporterWho	Specifies who is reporting
		AddresseeWho	Specifies the one to whom the Report is addressed
		ReportedWho	Specifies who is being reported on
	Requests		
		RequesterWho	Specifies who is making the request for some action
		RequestedWho	Specifies who is being requested to perform some action
Why			
	Orders		
		Why	Specifies reason for executing order
	Reports		
		ReporterWhy	Specifies the reason as perceived by the Reporter
		ObservedWhy	Specifies the reason as observed
When			
	Orders		
		OrderIssuedWhen	Specifies when the order was issued
		StartWhen	Start time of the mission
		EndWhen	End time of the mission
		RelativeWhen	A When that is relative to another When
	Reports		
		When	Specifies the time of the event in the Report
		WhenTime	Specifies the time
		WhenEvent	A form of When that defines time with respect to another Task
		WhenDetails	A detailed definition of a When
		RelativeWhen	A When that is relative to another When
Where			
	Orders, Reports and Requests		
		RouteWhere	Defines a route to be followed in action
		AtWhere	Defines Where an action is performed
		ControlFeatureWhere	A Where defined as a Control Feature
		StartWhere	A Where expressing an initial position
		EndWhere	A Where expressing a final position
What			
	Orders		
		What	Specifies the activity the tasked unit is to do
	Reports		
		ReporterWhat	Defines the perceived 'what' of the action
		ObservedWhat	Defines the observed 'what' of the action

Necessary distinctions in the usage of the 5Ws in specific expressions are indicated in the language constructs, as will be shown. The following subsection introduces the structural model for the language, describing the organization and content of the XML schemas that provide the information exchange content and structure specification, relating the information elements described in the XML schema to the underlying JC3IEDM data model.

Considerable additional information about organizations that are the subjects and/or objects of C-BML documents is required to initialize C2 and simulation systems for use with C-BML. The necessary information is specified under the MSDL standard. In order to ensure that systems receiving C-BML documents have access to the correct MSDL information, the C-BML schema contains an element describing the MSDL Scenario ID name, which is intended to serve as the source of such information as unit Tactical Graphics and Task Organization descriptions.

7.2 Structural Model of the Language

One formalism for describing the content and structure of C-BML expressions is the XML Schema language. This language provides a precise description of the information structure and content that can be used to *validate* (in the XML sense) XML documents containing C-BML expressions encoded in XML. XML documents are said to *validate* against an XML schema if XML validating parsers are able to determine that the documents correctly conform to the specification of information structure and content declared in the associated XML schema document (or documents, when declarations are organized into multiple schema files).

As introduced earlier in this specification, the use of XML facilitates widespread adoption and deployment of the C-BML standard given the ubiquitous use of XML in modern information systems. The Phase 1 C-BML XML representation of the 5Ws and associated constructs provides data types and information elements for use in expressing orders, requests, and reports that can be exchanged across systems through a variety of common information exchange mechanisms (messages, web services, etc.). Application of the approach for any specific service, organization, or nation requires transformation of respective information elements in current formats (e.g., textual or binary message formats), some of which may already use defined XML tagsets, into the C-BML XML structures. Legacy systems will require adapters to produce and consume C-BML expressions. Over time, however, as C-BML becomes widely adopted, it is likely that systems will emerge that natively “speak” C-BML, directly producing and processing C-BML expressions in place of older formats. Either way, systems will obtain the benefits of a shared, common structure and content for the expression of certain information elements in orders, requests, and reports.

XML schemas define how data are structured and described in XML documents. XML schemas define what values are valid (as defined above) for data contained in XML documents. XML schemas define what data types and structures can be used in construction of other schemas, enabling reuse of types and structures within and across application domains. An XML schema often declares an XML *namespace*, which acts as a label or container identifier for the names of data types and structures declared in the XML schema file. The use of namespaces allows software to distinguish between data types and structures that have the same names but are associated with different namespaces. Namespaces are generally identified by a string following the format of Universal Resource Names (URNs) or Universal Resource Locators (URLs).

Declarations of data types and structures in XML schemas can reference data types and structures that have been declared in other XML schemas. In this way, new schemas can reuse data types and structures defined in other schemas. Similarly, schemas for a particular application, such as C-BML, can be constructed in a modular fashion, organizing data types and structures to facilitate schema maintenance and use within that application or by other applications. XML schemas declare the use of another schema with the XML Schema language *include* statement, when the other schema is in the same namespace, or with the XML Schema language *import* statement, when the other schema has declared a different namespace.

In the following subsections, the organization of the XML schema files for the Phase 1 C-BML standard are described, followed by description of the data type and structure declarations in each C-BML XML schema file.

7.2.1 Phase 1 C-BML XML Schema Files

The Phase 1 C-BML XML schemas are organized into fourteen separate XML schema files to facilitate incremental evolution of the standard over time. The XML schema files are introduced below (note: file extension ".xsd" is a convention used to identify XML schema files)[2]:

- jc3iedm-codes.xsd (refer to Annex A)
    - Namespace: urn:int:nato:standard:mip:jc3iedm:3.0.2:oo:2.2
    - XML schema include: None
    - XML schema import: None
    - Purpose: This schema file declares enumeration data types extracted from the JC3IEDM XML schemas and provided in the Phase 1 C-BML distribution files (see Section 8) for referential convenience.
- jc3iedm-simple-types.xsd (refer to Annex B)
    - Namespace: urn:int:nato:standard:mip:jc3iedm:3.0.2:oo:2.2
    - XML schema include: jc3iedm-codes.xsd
    - XML schema import: None
    - Purpose: This schema file declares simple types (string, numeric, etc.) extracted from the JC3IEDM XML schemas and provided in the Phase 1 C-BML distribution files (see Section 8) for referential convenience.
- cbml-codes.xsd (refer to Annex C)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: None
    - XML schema import: None
    - Purpose: This schema file declares enumeration data types used in declarations in other C-BML schema files.
- cbml-entity-types.xsd (refer to Annex D)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-codes.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types relating to fundamental entities (in JC3IEDM parlance) and relationships across entities used in declarations in other C-BML schema files.
- cbml-composites.xsd (refer to Annex E)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-entity-types.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types and named information elements (XML structures) relating to the C-BML 5Ws (who, what, when, where, why) and their variations in usage in expression of orders, requests, and reports.

[2] The JC3IEDM namespace (urn:int:nato:standard:mip:jc3iedm:3.0.2:oo:2.2) uses the URN notation as is the convention in the MIP; the C-BML namespace (http://www.sisostds.org/schemas/c-bml/1.0) uses the URL notation.

- cbml-composites-light.xsd (refer to Annex F)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-composites.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares named information elements (XML structures) for constructing simplified C-BML expressions relating to description of tasks used in orders, requests, and reports.
- cbml-action-types.xsd (refer to Annex G)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-composites.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types and structures relating to actions and events.
- cbml-affiliation-types.xsd (refer to Annex H)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-entity-types.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types and structures relating to affiliation information (e.g., geopolitical, ethnic, functional, religion, other).
- cbml-facility-types.xsd (refer to Annex I)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-composites.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types relating to objects such as airfields, bridges, military obstacles, and roads.
- cbml-feature-types.xsd (refer to Annex J)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-entity-types.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types relating to objects such as geographic features (terrain characteristics to which military significance is attached), meteorological features (e.g., cloud cover, precipitation), and control features (e.g., route, airspace control means).
- cbml-location-types.xsd (refer to Annex K)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-entity-types.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types relating to positional information, including use of relative positioning and geometric shapes.

- cbml-materiel-types.xsd (refer to Annex L)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-entity-types.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types relating to items of materiel, such as aircraft, ammunition, vehicles, and weapons.
- cbml-organisation-types.xsd (refer to Annex M)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-entity-types.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types relating to organizations, such as military units, convoys, military posts, and task formations.
- cbml-person-types.xsd (refer to Annex N)
    - Namespace: http://www.sisostds.org/schemas/c-bml/1.0
    - XML schema include: cbml-composites.xsd
    - XML schema import: jc3iedm-simple-types.xsd
    - Purpose: This schema file declares data types relating to individual persons or classes of persons.

Figure 6 summarizes the XML schema dependency relationships identified above in the use of the *include* and *import* statements in the XML schema files. The diagram also shows which schemas are associated with which namespace. Note that all the schema files in the C-BML namespace *import* the jc3iedm-simple-types.xsd schema file.

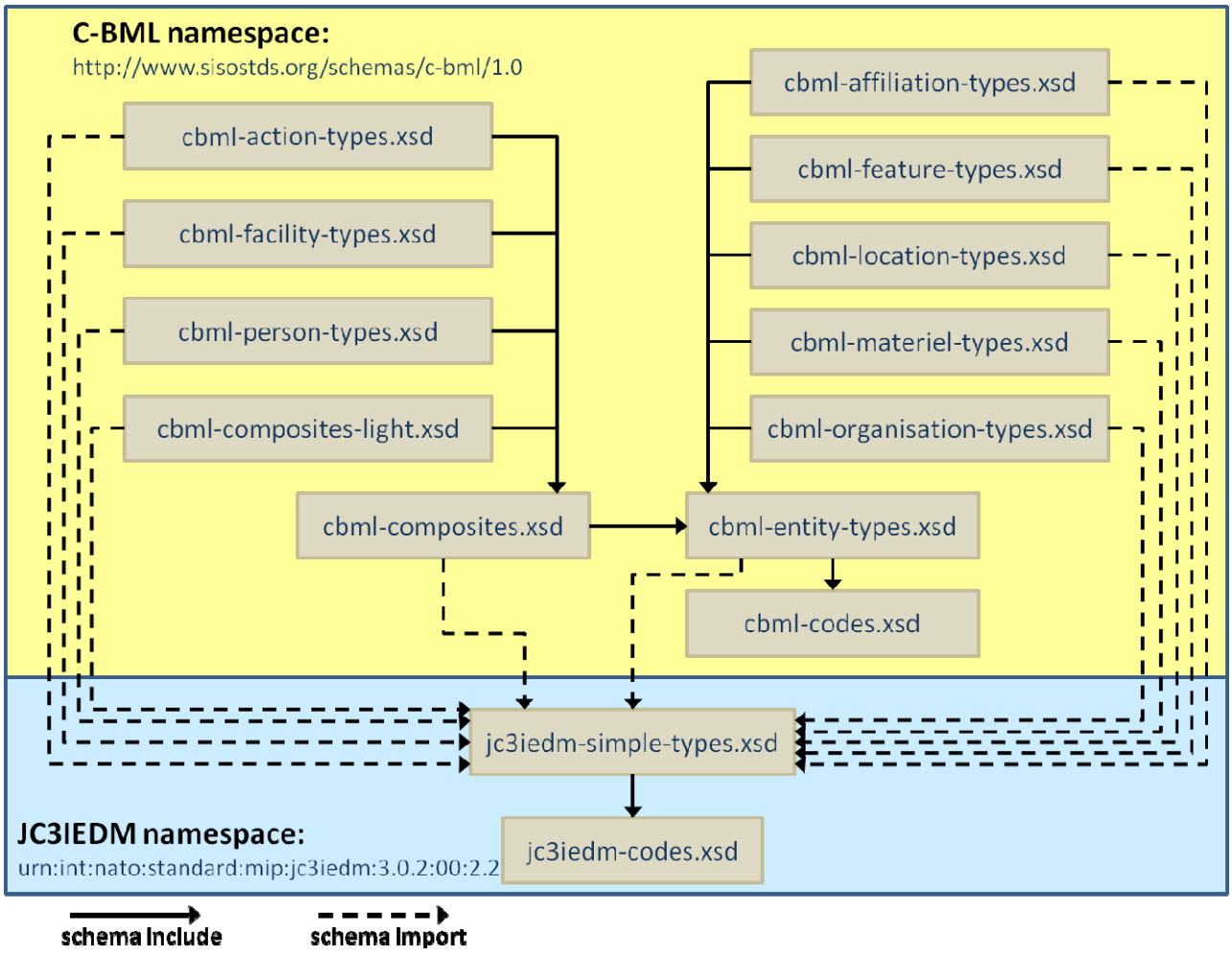

Figure 6: Phase 1 C-BML XML schema file dependencies

In the schema files, XML data types and named XML elements are declared at the global level to permit maximum flexibility in using the Phase 1 C-BML Specification. As shown above, the data types and elements are declared in a separate C-BML namespace, “http://www.sisostds.org/schemas/c-bml/1.0”. Users may include or import the C-BML constructs into their own XML schema designs to declare their own elements using C-BML data types or referencing C-BML named elements.

Subsections 7.2.2 through 7.2.7 to follow describe content of the Phase 1 C-BML XML schemas in more detail. For the full content of the XML schema files, refer to the respective Annexes identified above and reiterated in the following subsections.

7.2.2 JC3IEDM Reference Schemas: jc3iedm-codes.xsd and jc3iedm-simple-types.xsd

The **jc3iedm-codes.xsd** schema file declares enumeration data types extracted from the JC3IEDM XML schemas and provided in the Phase 1 C-BML distribution files (see Section 8) for referential convenience. This file also provides a place for C-BML developers and standards activity personnel to incrementally add new types or new enumerations to the existing types for evaluation prior to approval for new versions of the C-BML schemas and possible submission to the MIP.

Many enumeration data types (XML string-based simple types with restrictions on the allowable values) are declared in the jc3iedm-codes.xsd schema file; for example: AbsolutePointCategoryCode, ActionCategoryCode, ActionEffectCategoryCode, ActionTaskPriorityCode, and FacilityCategoryCode.

For example, the ActionCategoryCode enumeration simple type is declared as follows in the jc3iedm-codes.xsd schema:

```
<simpleType name="ActionCategoryCode">
  <annotation>
    <documentation xml:lang="en">The specific value that represents the class of
ACTION.</documentation>
  </annotation>
  <restriction base="token">
    <enumeration value="ACTEV">
      <annotation>
        <documentation>
          <Definition xml:lang="en">An ACTION that is an incident, phenomenon, or occasion of military
significance which has occurred or is occurring but for which planning is not known.</Definition>
        </documentation>
        <appinfo>
          <DisplayValue xml:lang="en">ACTION-EVENT</DisplayValue>
        </appinfo>
      </annotation>
    </enumeration>
    <enumeration value="ACTTA">
      <annotation>
        <documentation>
          <Definition xml:lang="en">An ACTION that is being or has been planned and for which the
planning details are known.</Definition>
        </documentation>
        <appinfo>
          <DisplayValue xml:lang="en">ACTION-TASK</DisplayValue>
        </appinfo>
      </annotation>
    </enumeration>
  </restriction>
</simpleType>
```

Refer to Annex A for the full schema and associated documentation for the jc3iedm-codes.xsd schema.

Similarly, the **jc3iedm-simple-types.xsd** schema file declares simple types (string, numeric, etc.) extracted from the JC3IEDM XML schemas and provided in the Phase 1 C-BML distribution files (see Section 8) for referential convenience. This file also provides a place for C-BML developers and standards activity personnel to incrementally add new simple types or modify existing types for evaluation prior to approval for new versions of the C-BML schemas and possible submission to the MIP.

Numerous simple types are declared in the jc3iedm-simple-types.xsd schema; for example: AngleTypeRangeAngle7_4, DatetimeTypeFix18, IdentifierType20, LatitudeCoordinateTypeRangeLatitude9_6, OIDType, and TemperatureTypeRangeTemperature5_1.

For example, the LatitudeCoordinateTypeRangeLatitude9_6 simple type is declared as follows in jc3iedm-simple-types.xsd schema:

```
<simpleType name="LatitudeCoordinateTypeRangeLatitude9_6">
  <annotation>
    <documentation xml:lang="en">The geodetic latitude of the location of a line or plane, expressed in
degrees, with positive values measured northward and negative values southward from the
equator.</documentation>
  </annotation>
  <restriction base="decimal">
    <totalDigits value="9"/>
```

```
    <fractionDigits value="6"/>
    <minInclusive value="-90.000000"/>
    <maxInclusive value="90.000000"/>
   </restriction>
  </simpleType>
```

Refer to Annex B for the full schema and associated documentation for the jc3iedm-simple-types.xsd schema.

7.2.3 C-BML Enumerations: cbml-codes.xsd

The **cbml-codes.xsd** schema file declares enumeration data types used in declarations in other C-BML schema files.

The following simple types are declared in the cbml-codes.xsd schema:

ActionEndTemporalAssociationCategoryCode
ActionStartTemporalAssociationCategoryCode
ActionTaskEndQualifierCode
ActionTaskEndTimeQualifierCode
ActionTaskStartQualifierCode
ActionTaskStartTimeQualifierCode
RequestCategoryCode

For example, the ActionEndTemporalAssociationCategoryCode enumeration simple type is declared as follows in the cbml-codes.xsd schema:

```
<xs:simpleType name="ActionEndTemporalAssociationCategoryCode">
 <xs:annotation>
  <xs:documentation xml:lang="en">The specific value that represents the class of chronological relationship of a subject ACTION to an object ACTION for a specific ACTION-TEMPORAL-ASSOCIATION.</xs:documentation>
 </xs:annotation>
 <xs:restriction base="xs:token">
  <xs:enumeration value="ENDEND">
   <xs:annotation>
    <xs:documentation>
     <xs:Definition xml:lang="en">The subject ACTION ends after the object ACTION ends.</xs:Definition>
    </xs:documentation>
    <xs:appinfo>
     <xs:DisplayValue xml:lang="en">Ends after end of</xs:DisplayValue>
    </xs:appinfo>
   </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="ENDENE">
   <xs:annotation>
    <xs:documentation>
     <xs:Definition xml:lang="en">The subject ACTION ends no earlier than the end of the object ACTION augmented by a fixed duration.</xs:Definition>
    </xs:documentation>
    <xs:appinfo>
     <xs:DisplayValue xml:lang="en">Ends no earlier than after end of</xs:DisplayValue>
    </xs:appinfo>
   </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="ENDENL">
   <xs:annotation>
    <xs:documentation>
```

```
        <xs:Definition xml:lang="en">The subject ACTION ends no later than the end of object
ACTION augmented by a fixed duration.</xs:Definition>
       </xs:documentation>
       <xs:appinfo>
        <xs:DisplayValue xml:lang="en">Ends no later than after end of</xs:DisplayValue>
       </xs:appinfo>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="ENDSNE">
     <xs:annotation>
      <xs:documentation>
       <xs:Definition xml:lang="en">The subject ACTION ends no earlier than the start of the object
ACTION augmented by a fixed duration.</xs:Definition>
      </xs:documentation>
      <xs:appinfo>
       <xs:DisplayValue xml:lang="en">Ends no earlier than after start of</xs:DisplayValue>
      </xs:appinfo>
     </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="ENDSNL">
     <xs:annotation>
      <xs:documentation>
       <xs:Definition xml:lang="en">The subject ACTION ends no later than the start of object
ACTION augmented by a fixed duration.</xs:Definition>
      </xs:documentation>
      <xs:appinfo>
       <xs:DisplayValue xml:lang="en">Ends no later than after start of</xs:DisplayValue>
      </xs:appinfo>
     </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="ENDSTR">
     <xs:annotation>
      <xs:documentation>
       <xs:Definition xml:lang="en">The subject ACTION ends after the object ACTION
starts.</xs:Definition>
      </xs:documentation>
      <xs:appinfo>
       <xs:DisplayValue xml:lang="en">Ends after start of</xs:DisplayValue>
      </xs:appinfo>
     </xs:annotation>
    </xs:enumeration>
   </xs:restriction>
</xs:simpleType>
```

Refer to Annex C for the full schema and associated documentation for the cbml-codes.xsd schema.

7.2.4 C-BML Entity Types: cbml-entity-types.xsd

The **cbml-entity-types.xsd** schema file declares data types relating to fundamental entities (in JC3IEDM parlance) and relationships across entities used in declarations in other C-BML schema files.

The following complex types are declared as *abstract* types in the cbml-entity-types.xsd schema:

AbstractAbsolutePoint
AbstractAbsolutePointRef
AbstractAction
AbstractActionEffect
AbstractActionEvent

AbstractActionEventRef
AbstractActionObjective
AbstractActionObjectiveItem
AbstractActionObjectiveItemRef
AbstractActionObjectiveRef
AbstractActionObjectiveType
AbstractActionObjectiveTypeRef
AbstractActionRef
AbstractActionResource
AbstractActionResourceEmployment
AbstractActionTask
AbstractActionTaskRef
AbstractAddress
AbstractAddressRef
AbstractAffiliation
AbstractAffiliationRef
AbstractCandidateTargetDetail
AbstractCandidateTargetDetailRef
AbstractCbrnEvent
AbstractCbrnEventRef
AbstractConsumableMaterielType
AbstractConsumableMaterielTypeRef
AbstractContext
AbstractContextRef
AbstractControlFeature
AbstractControlFeatureRef
AbstractControlFeatureType
AbstractControlFeatureTypeRef
AbstractEquipmentType
AbstractEquipmentTypeRef
AbstractFacility
AbstractFacilityRef
AbstractFacilityStatus
AbstractFacilityType
AbstractFacilityTypeRef
AbstractFeature
AbstractFeatureRef
AbstractFeatureType
AbstractFeatureTypeRef
AbstractGeographicFeatureStatus
AbstractGeometricVolume
AbstractGeometricVolumeRef
AbstractGovernmentOrganisationType
AbstractGovernmentOrganisationTypeRef
AbstractLocation
AbstractLocationRef
AbstractMateriel
AbstractMaterielRef
AbstractMaterielStatus
AbstractMaterielType
AbstractMaterielTypeRef
AbstractMeteorologicFeature
AbstractMeteorologicFeatureRef
AbstractMilitaryObstacle
AbstractMilitaryObstacleRef
AbstractMilitaryOrganisationType

AbstractMilitaryOrganisationTypeRef
AbstractMinefield
AbstractMinefieldRef
AbstractNuclearEvent
AbstractNuclearEventRef
AbstractObjectItem
AbstractObjectItemRef
AbstractObjectItemStatus
AbstractObjectType
AbstractObjectTypeRef
AbstractOrganisation
AbstractOrganisationRef
AbstractOrganisationType
AbstractOrganisationTypeRef
AbstractPoint
AbstractPointRef
AbstractRadioactiveEvent
AbstractRadioactiveEventRef
AbstractRelativeCoordinateSystem
AbstractRelativeCoordinateSystemRef
AbstractRouteSegment
AbstractRouteSegmentRef
AbstractSurface
AbstractSurfaceRef
AbstractVesselType
AbstractVesselTypeRef

An abstract type is used as a foundation for the declaration of other types but cannot be used to define an XML element that can be used directly by name in an XML instance document. For example, the AbstractAbsolutePoint complex type is declared as follows in the XML schema:

```
<xs:complexType name="AbstractAbsolutePoint" abstract="true">
  <xs:annotation>
    <xs:documentation xml:lang="en">A POINT in a geodetic system. Concrete types are:
{CartesianPoint, GeographicPoint}</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractPoint">
      <xs:sequence>
        <xs:choice minOccurs="0">
          <xs:element name="VerticalDistance" type="cbml:VerticalDistance">
            <xs:annotation>
              <xs:documentation xml:lang="en">The vertical distance for a specific ABSOLUTE-
POINT.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="VerticalDistanceRef" type="cbml:VerticalDistanceRef">
            <xs:annotation>
              <xs:documentation xml:lang="en">The vertical distance for a specific ABSOLUTE-
POINT.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:choice>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Note the use of the “abstract” attribute (set to value “true”) in the first line of this declaration. As noted in the description (annotation) of this complex type, concrete types derived from this abstract type include CartesianPoint and GeographicPoint. Both of these are declared in the cbml-location-types.xsd schema file (refer to the respective XML Schema declarations there). As derivations from this abstract complex type, both concrete types inherit the structure defined for the abstract type.

The following complex types (not abstract) are declared in the cbml-entity-types.xsd schema:

ActionContext
ActionContextInContext
ActionContextStatus
ActionEventStatus
ActionFunctionalAssociationInSubjectAction
ActionLocation
ActionObjectiveItemMarking
ActionObjectiveItemMarkingRef
ActionTaskStatus
CandidateTargetDetailItem
CandidateTargetDetailItemRef
CandidateTargetDetailType
CandidateTargetDetailTypeRef
CandidateTargetList
CandidateTargetListRef
Holding
HoldingRef
HoldingTransfer
ObjectItemAlias
ObjectItemAliasRef
ObjectItemAssociation
ObjectItemAssociationRef
ObjectItemAssociationStatus
ObjectItemLocation
ObjectItemLocationRef
ObjectItemObjectTypeEstablishment
ObjectItemObjectTypeEstablishmentInObjectItem
ObjectTypeEstablishment
ObjectTypeEstablishmentObjectTypeDetail
ObjectTypeEstablishmentObjectTypeDetailRef
ObjectTypeEstablishmentRef
OperationalInformationGroup
OperationalInformationGroupRef
OrganisationMaterielTypeAssociation
OrganisationMaterielTypeAssociationInOrganisation
OrganisationStructure
OrganisationStructureRef
OtherContext
OtherContextRef
OtherObjectItem
OtherObjectItemRef
OtherObjectItemStatus
OtherObjectType
OtherObjectTypeRef
SecurityClassification
SecurityClassificationRef
VerticalDistance
VerticalDistanceRef

The above complex types are sometimes called *concrete* types, since they can be used to define XML elements that can be used directly by name in an XML *instance* document (a document containing data values; in contrast to the *schema* document that only provides structural metadata to define how data are described in the instance document).

For example, the ObjectItemLocation complex type is declared as follows in the cbml-entity-types.xsd schema:

```
<xs:complexType name="ObjectItemLocation">
  <xs:annotation>
    <xs:documentation xml:lang="en">An association of an OBJECT-ITEM with a LOCATION that
enables the geographic position of the OBJECT-ITEM to be specified. The operational meaning of
geometry may also be specified.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="OID" type="jc3iedm:OIDType">
      <xs:annotation>
        <xs:documentation xml:lang="en">The globally unique object identifier. An OID can be any
globally unique string (URL, GUID...).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="VerticalAccuracyDimension" type="jc3iedm:DimensionType12_3"
minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The one-dimensional linear distance representing the
uncertainty in terms of probable error range for the vertical axis of a specific OBJECT-ITEM-
LOCATION.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="HorizontalAccuracyDimension" type="jc3iedm:DimensionType12_3"
minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The one-dimensional linear distance representing the
uncertainty in the horizontal plane in terms of probable circular error for a specific OBJECT-ITEM-
LOCATION.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="BearingAngle" type="jc3iedm:AngleTypeRangeAngle7_4" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The rotational measurement clockwise from the line of true
North to the direction of motion in the horizontal plane of a specific OBJECT-ITEM at a specific
LOCATION.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="BearingAccuracyAngle" type="jc3iedm:AngleTypeRangeAngle7_4"
minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The rotational measurement of a sector that represents the
uncertainty range in the estimate of the bearing at a specific OBJECT-ITEM-LOCATION. The sector
is bisected by the bearing.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="BearingPrecisionCode" type="jc3iedm:AnglePrecisionCode" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The specific value that represents the maximum resolution
used for the expression of a bearing angle.</xs:documentation>
      </xs:annotation>
```

```
    </xs:element>
    <xs:element name="InclinationAngle" type="jc3iedm:AngleTypeRangeAngle7_4" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The rotational measurement from the horizontal plane to the
direction of motion of a specific OBJECT-ITEM at a specific LOCATION (point or shape), where the
positive angle is vertically upward.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="InclinationAccuracyAngle" type="jc3iedm:AngleTypeRangeAngle7_4"
minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The rotational measurement of a vertical sector that
represents the uncertainty range in the estimate of the inclination at a specific OBJECT-ITEM-
LOCATION. The sector is bisected by the inclination.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="InclinationPrecisionCode" type="jc3iedm:AnglePrecisionCode"
minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The specific value that represents the maximum resolution
used for the expression of an inclination angle.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SpeedRate" type="jc3iedm:RateType8_4" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The numeric value that denotes the motion of a specific
OBJECT-ITEM at a specific LOCATION in terms of distance per unit time. The unit of measure is
kilometres per hour. The specified value must be greater than or equal to 0
(zero).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SpeedAccuracyRate" type="jc3iedm:RateType8_4" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The numeric value that denotes the uncertainty range in the
estimate for the speed at a specific OBJECT-ITEM-LOCATION where the speed estimate falls at the
centre of the accuracy range. The unit of measure is kilometres per hour. The specified value must be
greater than or equal to 0 (zero).</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SpeedPrecisionCode" type="jc3iedm:SpeedPrecisionCode" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The specific value that represents the maximum resolution
used for the expression of speed.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="MeaningCode" type="jc3iedm:ObjectItemLocationMeaningCode"
minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The specific value that represents the meaning of the
LOCATION geometry as it pertains to the OBJECT-ITEM.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="RelativeSpeedCode" type="jc3iedm:ObjectItemLocationRelativeSpeedCode"
minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">The specific value that represents the speed of the object
```

```
relative to its normal speed.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Like many of the other declarations in the C-BML schemas, this complex type is given a structure that parallels that defined in the XML schema representation of JC3IEDM produced by the MIP. Declaring the structure in the C-BML namespace enables it to be customized for C-BML applications. Changes to the structure by the C-BML community, managed through normal SISO processes, can be submitted to the MIP for consideration in new versions of JC3IEDM when the C-BML community considers it beneficial to all involved to do so.

Refer to Annex D for the full schema and associated documentation for the cbml-entity-types.xsd schema.

7.2.5 C-BML 5 Ws and Usage Distinctions: cbml-composites.xsd

The **cbml-composites.xsd** schema file declares data types and named information elements (XML structures) relating to the C-BML 5Ws (who, what, when, where, why) and their variations in usage in expression of orders, requests, and reports.

The following complex types are declared in the cbml-composites.xsd schema:

AbstractExpressionRefType
AbstractExpressionType
AbstractReportedWhenType
AbstractReportRefType
AbstractReportType
ActionType
AffectedWhoType
CandidateTargetListRefType
CandidateTargetListType
CoordinatingInformationType
EndWhenAbsoluteSpecifiedTimeType
EndWhenAbsoluteTimeType
EndWhenAbsoluteUnspecifiedTimeType
EndWhenRelativeTimeType
EventEndWhenType
EventStartWhenType
EventType
EventWhatLocationRefType
EventWhatLocationType
EventWhatRefType
EventWhatStatusRefType
EventWhatStatusType
EventWhatType
ExecuterWhoType
HoldingTransferRefType
HoldingTransferType

OrderTaskType
OrganisationStructureRefType
OrganisationStructureType
ReportedWhenAbsoluteTimingType
ReportedWhenRelativeTimingType
ReportedWhenType
ReporterWhoType
ReportingData

ReportingDataRef
RequestedWhoType
RequesterWhoType
RequestTaskType
RouteWhereType
SpecificLocationType
StartWhenAbsoluteSpecifiedTimeType
StartWhenAbsoluteTimeType
StartWhenAbsoluteUnspecifiedTimeType
StartWhenRelativeTimeType
StateType
TaskeeWhoType
TaskEndWhenType
TaskerWhoType
TaskLocalExtensionsType
TaskStartWhenType
TaskType
TaskWhatRefType
TaskWhatStatusRefType
TaskWhatStatusType
TaskWhatType
TaskWhenType
WhatEffectRefType
WhatEffectType
WhatRefType
WhatType
WhereType
WhoAffiliationRefType
WhoAffiliationType
WhoAssociationRefType
WhoAssociationType
WhoHoldingRefType
WhoHoldingType
WhoHostilityRefType
WhoHostilityType
WhoLocationRefType
WhoLocationType
WhoRefType
WhoStatusRefType
WhoStatusType
WhoType
WhoTypeRefType
WhoTypeType
WhyType

For example, the WhoType complex type is declared as follows in the cbml-composites.xsd schema:

```
<xs:complexType name="WhoType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Specifies a who.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ObjectItem" type="cbml:AbstractObjectItem">
      <xs:annotation>
        <xs:documentation xml:lang="en">An individually identified object that has military or civilian
significance. Concrete types are: {Atmosphere, CloudCover, Icing, Light, OtherMeteorologicFeature,
Precipitation, Visibility, Wind, AirRouteSegment, OtherRouteSegment, AirspaceControlMeans,
```

```
OtherControlFeature, Route, GeographicFeature, OtherFeature, InstrumentLandingSystem,
OtherMateriel, MinefieldLand, MinefieldMaritime, OtherMilitaryObstacle, Airfield, Anchorage, Apron,
Basin, Berth, Bridge, DryDock, Harbour, Jetty, OtherFacility, Quay, Railway, Road, Slipway, Convoy,
OtherOrganisation, Unit, OtherObjectItem, Person}</xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:sequence>
</xs:complexType>
```

The following XML elements (named data structures that can be used directly by name in XML instance documents) are declared in the cbml-composites.xsd schema:

AffectedWho
CoordinatingInformation
EventEndWhen
EventStartWhen
EventWhat
EventWhatRef
ExecuterWho
ReportedWhen
ReporterWho
RequestedWho
RequesterWho
RouteWhere
TaskeeWho
TaskerWho
TaskWhat
TaskWhatRef
TaskWhen
What
WhatRef
Where
Who
WhoRef
Why

For example, the Who element is declared as follows in the cbml-composites.xsd schema, referencing the respective complex type defined in the schema file:

```
<xs:element name="Who" type="cbml:WhoType"/>
```

The C-BML 5Ws define information elements that can be employed to construct a variety of expressions. This is important because the C-BML Phase 1 Specification is not based on any particular doctrine and thus does not mandate the structure and content of a specific set of expressions. Rather, this Phase 1 C-BML Specification defines basic information components that are found in many different doctrinal expressions. Early adopters can employ the data types and XML elements in their own XML schemas to create any structures they wish by invoking the applicable XML schema and declaring the C-BML namespace. The context and requirements of any specific information exchange will dictate what data types and elements the community requires in order for the exchange to function correctly. However, once an element in the top level is selected, users have to conform to the requirements and constraints defined by the Phase 1 C-BML schemas. As an example, consider the C-BML “Who” element shown above. The schema excerpt indicates that a “Who” can be any of the concrete types (Unit, Road, Bridge, etc.). There is no mandate that a “Who” be any of those objects; however, if systems want to exchange information about a Unit in particular, they must follow the structure defined by the schema for that data type (defined in the cbml-organisation-types.xsd schema). The Phase 1 C-BML Guidelines document provides examples of the construction of custom C-BML expressions from the information components defined in the Phase 1 C-BML XML schemas. The work of early adopters of the Phase 1 C-BML standard

to identify and define the structure and content of C-BML expressions needed for their applications will provide valuable information for development of the formal grammar in the Phase 2 C-BML effort.

Note that the TaskType structure includes an optional child element, TaskLocalExtensions, that can contain any free text information deemed useful by the C-BML community.

Considerable additional information about organizations that are the subjects and/or objects of C-BML documents is required to initialize C2 and simulation systems for use with C-BML. The necessary information is specified under the MSDL standard. In order to ensure that systems receiving C-BML documents have access to the correct MSDL information, the C-BML schema contains an element (CoordinatingInformation) describing the MSDL Scenario ID name, which is intended to serve as the source of such information as unit Tactical Graphics and Task Organization descriptions.

Refer to Annex E for the full schema and associated documentation for the cbml-composites.xsd schema.

7.2.6 Initial C-BML Expression Structure: cbml-composites-light.xsd

The **cbml-composites-light.xsd** schema file declares data types and named information elements (XML structures) for constructing an initial set of C-BML expressions relating to description of tasks used in orders, requests, and reports. The “light” constructs declared in this schema file and in the cbml-composites.xsd schema file are intended to provide a simple set of information elements, considered adequate for many purposes, that limits the use of XML abstract types in favor of XML “choice” structures (where the choice is a selection from a number of concrete types).

The following complex types are declared in the cbml-composites-light.xsd schema:

AbsoluteReportedWhenLightType
AffectedWhoLightType
AtWhereLightType
CorridorAreaLightType
DesiredEffectLightType
DisplacementLightType
EndWhenRelativeTimeLightType
GDCLightType
LineLightType
LocationLightType
PointLightType
RelativeReportedWhenLightType
ReportHeaderLightType
RouteLightType
RouteWhereLightType
SpecificPointLightType
StartWhenRelativeTimeLightType
SupportedTaskLightType
SurfaceLightType
TaskEndWhenLightType
TaskLightType
TaskStartWhenLightType
TaskWhenLightType
TaskWhyLightType

For example, the TaskLightType complex type is declared as follows in the cbml-composites-light.xsd schema, using several data types declared in other Phase 1 C-BML XML schemas:

```
<xs:complexType name="TaskLightType">
 <xs:annotation>
  <xs:documentation xml:lang="en"> Specifies the composites making up a task. Necessary to
        describe Tasks in Orders and Requests.</xs:documentation>
 </xs:annotation>
 <xs:sequence>
```

```
    <xs:element name="TaskID" type="jc3iedm:OIDType">
      <xs:annotation>
        <xs:documentation> ID to be assigned to this task. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TaskeeWhoRef" type="jc3iedm:OIDType">
      <xs:annotation>
        <xs:documentation> Who that is to carry out the task. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ActivityCode" type="jc3iedm:ActionTaskActivityCode">
      <xs:annotation>
        <xs:documentation> What is to be done and when the task is to begin and end.
                </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TaskWhenLight" type="cbml:TaskWhenLightType">
      <xs:annotation>
        <xs:documentation> What is to be done and when the task is to begin and end.
                </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:choice>
      <xs:element name="AtWhere" type="cbml:AtWhereLightType">
        <xs:annotation>
          <xs:documentation> The location of or the route to be followed in carrying
                    out the task. </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="RouteWhereLight" type="cbml:RouteWhereLightType"/>
    </xs:choice>
    <xs:element name="AffectedWhoLight" type="cbml:AffectedWhoLightType" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation> Units or Objects who may be affected by the task.
                </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="WhyLight" type="cbml:TaskWhyLightType" minOccurs="0">
      <xs:annotation>
        <xs:documentation> Why the task is being carried out. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TaskControlMeasures" type="jc3iedm:OIDType" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation> Geographic constraints on the task. </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TaskLocalExtensions" type="cbml:TaskLocalExtensionsType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Local extensions that are not part of the standard should be
added here.</xs:documentation>
      </xs:annotation>
    </xs:element>
```

```
  </xs:sequence>
</xs:complexType>
```

For example, an element declaration for a Task can be declared using the above complex type::

```
<xs:element name="Task" type="cbml:TaskLightType"/>
```

The Task structure provided by the TaskLightType complex type is pictured in Figure 7. This structure has been used successfully in several previous prototype and experimentation efforts, and is provided in the Phase 1 C-BML XML schemas as a structure that can be employed “out-of-the-box” by early adopters of the Phase 1 standard, when that structure is applicable and sufficient for users’ needs.

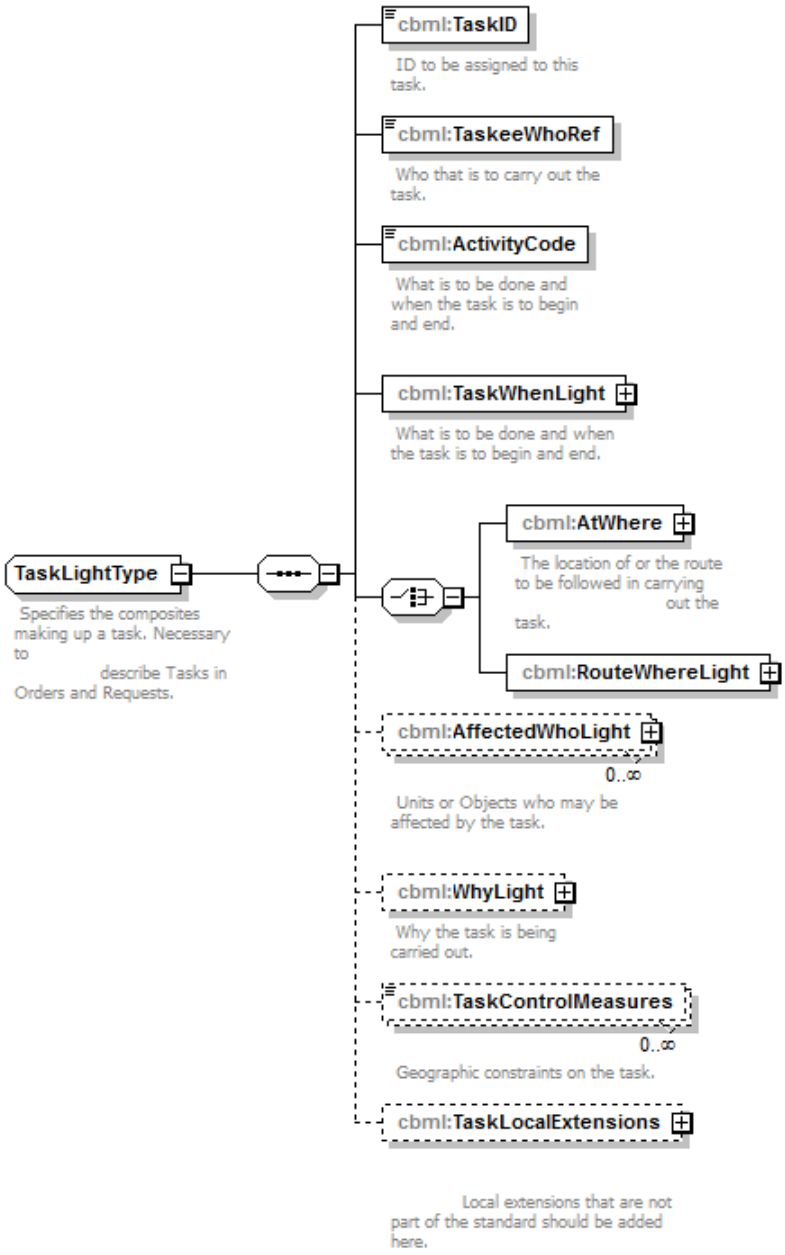

Figure 7: Phase 1 C-BML TaskLightType XML structure

Refer to Annex F for the full schema and associated documentation for the cbml-composites-light.xsd schema.

7.2.7 C-BML Concrete Types: Action, Affiliation, Facility, Feature, Location, Materiel, Organisation, Person

The eight Phase 1 C-BML XML schemas described in this subsection declare concrete types derived from abstract data types defined in other schema files. Examples extracted from the XML schemas demonstrate how declarations of complex types as extensions of abstract types simplify the declarations, whereby the concrete types inherit the content and structure of the abstract type.

The **cbml-action-types.xsd** schema file declares data types relating to actions and events. The following complex types are declared in the cbml-action-types.xsd schema:

ActionEffectItem
ActionEffectType
ActionObjectiveTask
ActionObjectiveTaskRef
ActionResourceItem
ActionResourceType
ChemicalBiologicalEvent
ChemicalBiologicalEventRef
NuclearWeaponEvent
NuclearWeaponEventRef
OtherActionEvent
OtherActionEventRef
OtherActionObjectiveItem
OtherActionObjectiveItemRef
OtherActionObjectiveType
OtherActionObjectiveTypeRef
OtherActionResourceEmployment
OtherActionTask
OtherActionTaskRef
OtherCbrnEvent
OtherCbrnEventRef
OtherNuclearEvent
OtherNuclearEventRef
OtherRadioactiveEvent
OtherRadioactiveEventRef
RadiologicalEvent
RadiologicalEventRef
Request
RequestAnswer
RequestRef
RequestWhatAnswerRefType
RequestWhatAnswerType
Target
TargetPersonnelProtectionRefType
TargetPersonnelProtectionType
TargetRef

For example, the ActionEffectItem complex type is declared as follows in the cbml-action-types.xsd schema:

```
<xs:complexType name="ActionEffectItem">
  <xs:annotation>
    <xs:documentation xml:lang="en">An ACTION-EFFECT of a specific ACTION in accomplishing its
aim in relation to a specific OBJECT-ITEM.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractActionEffect">
```

```
      <xs:sequence>
        <xs:element name="Ratio" type="jc3iedm:RatioType6_5">
          <xs:annotation>
            <xs:documentation xml:lang="en">The numeric quotient value that represents the portion of a
whole OBJECT-ITEM that is estimated in a specific ACTION-EFFECT-ITEM to have the result
specified in ACTION-EFFECT. The value must be in the range from 0 to 1.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="ObjectItemRef" type="cbml:AbstractObjectItemRef">
          <xs:annotation>
            <xs:documentation xml:lang="en">Reference to an OBJECT-ITEM. Concrete types are:
{AtmosphereRef, CloudCoverRef, IcingRef, LightRef, OtherMeteorologicFeatureRef, PrecipitationRef,
VisibilityRef, WindRef, AirRouteSegmentRef, OtherRouteSegmentRef, AirspaceControlMeansRef,
OtherControlFeatureRef, RouteRef, GeographicFeatureRef, OtherFeatureRef,
InstrumentLandingSystemRef, OtherMaterielRef, MinefieldLandRef, MinefieldMaritimeRef,
OtherMilitaryObstacleRef, AirfieldRef, AnchorageRef, ApronRef, BasinRef, BerthRef, BridgeRef,
DryDockRef, HarbourRef, JettyRef, OtherFacilityRef, QuayRef, RailwayRef, RoadRef, SlipwayRef,
ConvoyRef, OtherOrganisationRef, UnitRef, OtherObjectItemRef, PersonRef}</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Refer to Annex G for the full schema and associated documentation for the cbml-action-types.xsd schema.

The **cbml-affiliation-types.xsd** schema file declares data types relating to affiliation information (e.g., geopolitical, ethnic, functional, religion, other). The following complex types are declared in the cbml-affiliation-types.xsd schema:

AffiliationEthnicGroup
AffiliationEthnicGroupRef
AffiliationFunctionalGroup
AffiliationFunctionalGroupRef
AffiliationGeopolitical
AffiliationGeopoliticalRef
AffiliationReligion
AffiliationReligionRef
OtherAffiliation
OtherAffiliationRef

For example, the AffiliationGeopolitical complex type is declared as follows in the cbml-affiliation-types.xsd schema:

```
<xs:complexType name="AffiliationGeopolitical">
  <xs:annotation>
    <xs:documentation xml:lang="en">A specification of a country or political entity to which
membership or allegiance may be ascribed.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractAffiliation">
      <xs:sequence>
        <xs:element name="AffiliationGeopoliticalCode" type="jc3iedm:AffiliationGeopoliticalCode">
          <xs:annotation>
            <xs:documentation xml:lang="en">The specific value that represents the identification of the
independent first-level geographic-political area and its dependencies, areas of quasi-independence,
```

```
and areas with special unrecognised sovereignty, including outlying and disputed
areas.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

Refer to Annex H for the full schema and associated documentation for the cbml-affiliation-types.xsd schema.

The **cbml-facility-types.xsd** schema file declares data types relating to objects such as airfields, bridges, military obstacles, and roads. The following complex types are declared in the cbml-facility-types.xsd schema:

Airfield
AirfieldRef
AirfieldStatus
AirfieldType
AirfieldTypeRef
Anchorage
AnchorageRef
ApproachDirection
ApproachDirectionRef
Apron
ApronRef
Basin
BasinRef
Berth
BerthRef
Bridge
BridgeRef
BridgeType
BridgeTypeRef
DryDock
DryDockRef
ElectronicAddress
ElectronicAddressRef
Harbour
HarbourRef
HarbourType
HarbourTypeRef
Jetty
JettyRef
MedicalFacilityStatus
MedicalFacilityStatusCasualtyBedOccupancy
MedicalFacilityStatusCasualtyBedOccupancyRef
MedicalFacilityStatusIntervalCasualtyGroup
MedicalFacilityStatusIntervalCasualtyGroupRef
MedicalFacilityStatusIntervalCasualtyType
MedicalFacilityStatusIntervalCasualtyTypeRef
MedicalFacilityStatusIntervalEvacuation
MedicalFacilityStatusIntervalEvacuationRef
MedicalFacilityStatusPendingCasualtyEvacuation
MedicalFacilityStatusPendingCasualtyEvacuationRef
MedicalFacilityStatusPendingSurgery

MedicalFacilityStatusPendingSurgeryRef
MilitaryObstacleType
MilitaryObstacleTypeRef
MinefieldLand
MinefieldLandRef
MinefieldMaritime
MinefieldMaritimeCasualtyEstimate
MinefieldMaritimeCasualtyEstimateRef
MinefieldMaritimeRef
MinefieldMaritimeStatus
MinefieldMaritimeSustainedThreatMeasureOfEffectiveness
MinefieldMaritimeSustainedThreatMeasureOfEffectivenessRef
Network
NetworkCapacity
NetworkCapacityRef
NetworkFrequency
NetworkFrequencyRef
NetworkRef
NetworkService
NetworkServiceRef
NetworkServiceStatusRefType
NetworkServiceStatusType
ObjectItemAddress
OtherAddress
OtherAddressRef
OtherFacility
OtherFacilityRef
OtherFacilityStatus
OtherFacilityType
OtherFacilityTypeRef
OtherMilitaryObstacle
OtherMilitaryObstacleRef
Quay
QuayRef
Railway
RailwayRef
Road
RoadRef
Runway
RunwayApproachDirectionAssociation
RunwayApproachDirectionAssociationInRunway
RunwayApproachDirectionAssociationRef
RunwayRef
Slipway
SlipwayRef
WhoAddressRefType
WhoAddressType

For example, the Road complex type is declared as follows in the cbml-facility-types.xsd schema:

```
<xs:complexType name="Road">
  <xs:annotation>
    <xs:documentation xml:lang="en">A FACILITY that is a path or way with a specially prepared
surface.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractFacility">
```

```
    <xs:sequence>
      <xs:element name="RoadCategoryCode" type="jc3iedm:RoadCategoryCode">
        <xs:annotation>
          <xs:documentation xml:lang="en">The specific value that represents the type of
ROAD.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ShoulderWidthCode" type="jc3iedm:RoadShoulderWidthCode"
minOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">The specific value that represents the average horizontal
distance measured from side to side and perpendicular to the central axis of a specific hard shoulder
(lane/area beside a highway for broken-down or not running vehicles).</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="TrafficDensityCount" type="jc3iedm:CountType4" minOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">The integer value representing the average number of
vehicles that occupy one kilometre of road space, expressed in vehicles per
kilometre.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="WeatherConditionCategoryCode"
type="jc3iedm:RoadWeatherConditionCategoryCode" minOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">The specific value that describes the passability of a ROAD
considering the impact of weather on that ROAD.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="QualityCode" type="jc3iedm:RoadQualityCode" minOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">The specific value that represents a subjective rating of the
quality of the ROAD.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

Refer to Annex I for the full schema and associated documentation for the cbml-facility-types.xsd schema.

The **cbml-feature-types.xsd** schema file declares data types relating to objects such as geographic features (terrain characteristics to which military significance is attached), meteorological features (e.g., cloud cover, precipitation), and control features (e.g., route, airspace control means). The following complex types are declared in the cbml-feature-types.xsd schema:

AirRouteSegment
AirRouteSegmentRef
AirspaceControlMeans
AirspaceControlMeansRef
Atmosphere
AtmosphereRef
CloudCover
CloudCoverRef
ControlFeatureStatus
GeographicFeature

GeographicFeatureRef
GeographicFeatureType
GeographicFeatureTypeRef
Icing
IcingRef
Light
LightRef
LiquidBodyStatus
LiquidSurfaceStatus
OtherControlFeature
OtherControlFeatureRef
OtherControlFeatureType
OtherControlFeatureTypeRef
OtherFeature
OtherFeatureRef
OtherFeatureType
OtherFeatureTypeRef
OtherMeteorologicFeature
OtherMeteorologicFeatureRef
OtherRouteSegment
OtherRouteSegmentRef
Precipitation
PrecipitationRef
Route
RouteRef
RouteType
RouteTypeRef
SolidSurfaceStatus
Visibility
VisibilityRef
Wind
WindRef

For example, the Route complex type is declared as follows in the cbml-feature-types.xsd schema:

```
<xs:complexType name="Route">
  <xs:annotation>
    <xs:documentation xml:lang="en">A CONTROL-FEATURE that is the prescribed course to be
travelled from a specific point of origin to a specific destination.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractControlFeature">
      <xs:sequence>
        <xs:element name="DirectionUsageCode" type="jc3iedm:RouteDirectionUsageCode"
minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The specific value that represents the assigned direction
for the traffic on the route.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="MobilityCode" type="jc3iedm:MobilityCode" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The specific value that indicates the suitability of a specific
ROUTE for movement.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="ModeOfTransportationCode" type="jc3iedm:ModeOfTransportationCode"
```

```
minOccurs="0">
        <xs:annotation>
          <xs:documentation xml:lang="en">The specific value that indicates the mode of transportation
of a specific ROUTE.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

Refer to Annex J for the full schema and associated documentation for the cbml-feature-types.xsd schema.

The **cbml-location-types.xsd** schema file declares data types relating to positional information, including use of relative and absolute locations. The following complex types are declared in the cbml-location-types.xsd schema:

CartesianPoint
CartesianPointRef
ConeVolume
ConeVolumeRef
CorridorArea
CorridorAreaRef
Ellipse
EllipseRef
FanArea
FanAreaRef
GeographicPoint
GeographicPointRef
Line
LineRef
ObjectReference
ObjectReferenceRef
OrbitArea
OrbitAreaRef
OtherLocation
OtherLocationRef
PointReference
PointReferenceRef
PolyarcArea
PolyarcAreaRef
PolygonArea
PolygonAreaRef
RelativePoint
RelativePointRef
SphereVolume
SphereVolumeRef
SurfaceVolume
SurfaceVolumeRef
TrackArea
TrackAreaRef

For example, the GeographicPoint complex type is declared as follows in the cbml-location-types.xsd schema:

```
<xs:complexType name="GeographicPoint">
 <xs:annotation>
```

```
    <xs:documentation xml:lang="en">An ABSOLUTE-POINT that has its position specified with
respect to the surface of the World Geodetic System 1984 (WGS 84) ellipsoid.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractAbsolutePoint">
      <xs:sequence>
        <xs:element name="LatitudeCoordinate"
type="jc3iedm:LatitudeCoordinateTypeRangeLatitude9_6">
          <xs:annotation>
            <xs:documentation xml:lang="en">The numeric value that represents the angle between the
plane of the equator and a line perpendicular to the ellipsoid surface and passing through the
GEOGRAPHIC-POINT.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="LongitudeCoordinate"
type="jc3iedm:LongitudeCoordinateTypeRangeLongitude10_6">
          <xs:annotation>
            <xs:documentation xml:lang="en">The numeric value that represents the angle between the
initial (zero or Greenwich) meridian and the meridian of the GEOGRAPHIC-POINT measured in the
plane of the Equator.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="LatitudePrecisionCode" type="jc3iedm:AnglePrecisionCode"
minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The specific value that represents the resolution used for
the expression of a value of a latitude coordinate.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="LongitudePrecisionCode" type="jc3iedm:AnglePrecisionCode"
minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The specific value that represents the resolution used for
the expression of a value of a longitude coordinate.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Refer to Annex K for the full schema and associated documentation for the cbml-location-types.xsd schema.

The **cbml-materiel-types.xsd** schema file declares data types relating to items of materiel, such as aircraft, ammunition, vehicles, and weapons. The following complex types are declared in the cbml-materiel-types.xsd schema:

AircraftType
AircraftTypeRef
AmmunitionType
AmmunitionTypeRef
BiologicalMaterielType
BiologicalMaterielTypeRef
CbrnEquipmentType
CbrnEquipmentTypeRef
ChemicalMaterielType

ChemicalMaterielTypeRef
ElectronicEquipmentType
ElectronicEquipmentTypeRef
EngineeringEquipmentType
EngineeringEquipmentTypeRef
InstrumentLandingSystem
InstrumentLandingSystemRef
MaritimeEquipmentType
MaritimeEquipmentTypeRef
MineStatus
MiscellaneousEquipmentType
MiscellaneousEquipmentTypeRef
OtherConsumableMaterielType
OtherConsumableMaterielTypeRef
OtherMateriel
OtherMaterielRef
OtherMaterielStatus
OtherMaterielType
OtherMaterielTypeRef
OtherVesselType
OtherVesselTypeRef
RadioactiveMaterielType
RadioactiveMaterielTypeRef
RailcarType
RailcarTypeRef
SubsurfaceVesselType
SubsurfaceVesselTypeRef
SurfaceVesselType
SurfaceVesselTypeRef
UxoStatus
VehicleType
VehicleTypeRef
WeaponType
WeaponTypeRef

For example, the VehicleType complex type is declared as follows in the cbml-materiel-types.xsd schema:

```
<xs:complexType name="VehicleType">
  <xs:annotation>
    <xs:documentation xml:lang="en">An EQUIPMENT-TYPE that is designed to operate on land routes (other than rail) with a primary role of transporting personnel, equipment or supplies.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractEquipmentType">
      <xs:sequence>
        <xs:element name="VehicleTypeCategoryCode" type="jc3iedm:VehicleTypeCategoryCode">
          <xs:annotation>
            <xs:documentation xml:lang="en">The specific value that represents the class of VEHICLE-TYPE.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Refer to Annex L for the full schema and associated documentation for the cbml-materiel-types.xsd schema.

The **cbml-organisation-types.xsd** schema file declares data types relating to organizations, such as military units, convoys, military posts, and task formations. The following complex types are declared in the cbml-organisation-types.xsd schema:

CivilianPostType
CivilianPostTypeRef
Convoy
ConvoyRef
ExecutiveMilitaryOrganisationType
ExecutiveMilitaryOrganisationTypeRef
GroupOrganisationType
GroupOrganisationTypeRef
MilitaryPostType
MilitaryPostTypeRef
OrganisationStatus
OtherGovernmentOrganisationType
OtherGovernmentOrganisationTypeRef
OtherMilitaryOrganisationType
OtherMilitaryOrganisationTypeRef
OtherOrganisation
OtherOrganisationRef
OtherOrganisationType
OtherOrganisationTypeRef
PrivateSectorOrganisationType
PrivateSectorOrganisationTypeRef
TaskFormationType
TaskFormationTypeRef
Unit
UnitRef
UnitType
UnitTypeRef

For example, the Unit complex type is declared as follows in the cbml-organisation-types.xsd schema:

```
<xs:complexType name="Unit">
  <xs:annotation>
    <xs:documentation xml:lang="en">A military ORGANISATION whose structure is prescribed by
competent authority.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractOrganisation">
      <xs:sequence>
        <xs:element name="FormalAbbreviatedNameText" type="jc3iedm:TextTypeVar100">
          <xs:annotation>
            <xs:documentation xml:lang="en">The character string specifying the common formal
abbreviation used to designate a specific UNIT.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="IdentificationText" type="jc3iedm:TextTypeVar15" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The character string assigned to represent a unit's
identification.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
```

```
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Refer to Annex M for the full schema and associated documentation for the cbml-organisation-types.xsd schema.

The **cbml-person-types.xsd** schema file declares data types relating to individual persons or classes of persons. The following complex types are declared in the cbml-person-types.xsd schema:

GroupCharacteristic
GroupCharacteristicRef
ObjectItemGroupAccount
ObjectItemGroupAccountDetail
ObjectItemGroupAccountDetailRef
Person
PersonIdentificationDocument
PersonIdentificationDocumentRef
PersonLanguageSkill
PersonLanguageSkillRef
PersonRef
PersonStatus
PersonType
PersonTypeRef
WhoGroupAccountRefType
WhoGroupAccountType

For example, the Person complex type is declared as follows in the cbml-person-types.xsd schema:

```
<xs:complexType name="Person">
  <xs:annotation>
    <xs:documentation xml:lang="en">An OBJECT-ITEM that is a human being to whom military or
civilian significance is attached.</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="cbml:AbstractObjectItem">
      <xs:sequence>
        <xs:element name="BirthDatetime" type="jc3iedm:DatetimeTypeFix18" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">The character string representing a point in time that
designates the date when a specific PERSON was born.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="BloodTypeCode" type="jc3iedm:PersonBloodTypeCode" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">A code which represents the specific blood type of a
PERSON.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="GenderCode" type="jc3iedm:GenderCode" minOccurs="0">
          <xs:annotation>
            <xs:documentation xml:lang="en">A code that represents the classification of a PERSON
based on reproductive physiological traits.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="ProfessingIndicatorCode" type="jc3iedm:PersonProfessingIndicatorCode"
minOccurs="0">
          <xs:annotation>
```

```
        <xs:documentation xml:lang="en">The specific value that represents whether a specific
PERSON professes a religious preference.</xs:documentation>
       </xs:annotation>
      </xs:element>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:element name="IdentificationDocument" type="cbml:PersonIdentificationDocument">
        <xs:annotation>
         <xs:documentation xml:lang="en">A child in a 'is-identified-by'
relationship.</xs:documentation>
        </xs:annotation>
       </xs:element>
       <xs:element name="IdentificationDocumentRef"
type="cbml:PersonIdentificationDocumentRef">
        <xs:annotation>
         <xs:documentation xml:lang="en">A child in a 'is-identified-by'
relationship.</xs:documentation>
        </xs:annotation>
       </xs:element>
      </xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:element name="LanguageSkill" type="cbml:PersonLanguageSkill">
        <xs:annotation>
         <xs:documentation xml:lang="en">A child in a 'is-recognised-as-having'
relationship.</xs:documentation>
        </xs:annotation>
       </xs:element>
       <xs:element name="LanguageSkillRef" type="cbml:PersonLanguageSkillRef">
        <xs:annotation>
         <xs:documentation xml:lang="en">A child in a 'is-recognised-as-having'
relationship.</xs:documentation>
        </xs:annotation>
       </xs:element>
      </xs:choice>
     </xs:sequence>
    </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Refer to Annex N for the full schema and associated documentation for the cbml-person-types.xsd schema.

7.3 Phase 1 C-BML XML Schema Constructs and JC3IEDM Implementations

The Phase 1 C-BML schemas use the logical view of the JC3IEDM [19] as made available by the MIP in XML schema files [20]. The C-BML schemas convey logical dependencies of the constructs; physical implementation is left to the user. This is a deliberate choice made to allow for a flexible implementation of the standard while maintaining and enforcing a common high level (logical) view across all C-BML compliant systems. In early stages of adoption, some C-BML users will use legacy systems that they will turn into a C-BML compliant system. Such users need not throw away existing solutions which are at the physical level (already implemented), but can generate a logical view of their system that can fulfill the information exchange requirements mandated by the C-BML logical view. As an example, consider two systems A and B that need to exchange initial information about a Unit using the C-BML “Who” schema. The schemas indicate that a Unit has to have a name, a type, and a formal abbreviated name. While at the the logical level this information is sufficient, it is ambiguous when one has to turn it into a physical item. As a result, additional requirements found in the JC3IEDM business rules must be applied. In JC3IEDM, business rules specify constraints that either cannot be expressed in formal IDEF1X notation or those that are not explicitly structured as a design choice [16]. In this case, the user has to at least

unambiguously identify the unit once using a globally unique identifier. In addition, it is mandated that a unit type be defined before that type is referenced; thus, the physical implementation has to implement a way to insure that this mandate is respected. Consequently, the following algorithm can be employed to transition from the logical view to the physical view for this situation:

1. Check that the unit type being referenced exists.
2. If yes, continue to 3; if not, reject the information as incomplete.
3. Check the unique identifier. If the object already exists, accept the information and move to 5.
4. If the object does not exist, create the object and assign a unique identifier.
5. Exit

The implementation of this algorithm is left to the user. Many systems have been developed using a relational database approach; however, the Phase 1 C-BML standard does not advocate one approach over another as long as the information exchange requirements are fullfiled. The close coupling of the C-BML schemas to the JC3IEDM is a direct result of the decision to employ the JC3IEDM as the underlying data model for the Phase 1 C-BML standard. Introduction of other data models will require a mapping from the C-BML information components to the new target data models, either directly or by retaining JC3IEDM as an intermediate structure. Follow-on C-BML development phases may introduce new approaches.

Use of the C-BML schema shall adhere to all of the business rules specified by the MIP [16]. Users are encouraged to familiarize themselves with those rules in order to accurately use C-BML. Physical implementations shall remain consistent with the logical schemas and not add additional constraints that are not defined or enforced by the schemas. All mandatory elements in the schemas shall remain mandatory, and no optional element shall become mandatory when transitioning from the logical to the physical view and vice versa.

Use of the JC3IEDM namespace and declaration of C-BML data types that parallel JC3IEDM data structures directly tie specification of the Phase 1 C-BML constructs to the underlying data model.

7.4 Compliance to the Phase 1 C-BML Specification

Interoperability is the ability of two or more systems or components to exchange information and the ability of those systems to use the information that has been exchanged [B14]. Definition of compliance to the Phase 1 C-BML specification provides the C-BML community strong assurance that a system will be able to interoperate with all other compliant systems. To comply with this C-BML specification, systems shall generate, exchange, and process valid C-BML expressions, where validity is based on fulfillment of the structure and content requirements as specified in the C-BML XML schemas. Employment of this specification includes: (1) direct use of the constructs in the C-BML schemas to construct C-BML expressions and (2) indirect use of the constructs in the C-BML schemas in user-defined schemas developed to specify expressions not explicitly defined by the Phase 1 C-BML schemas. In the former case, compliant C-BML expressions shall validate (in the XML sense) against the Phase 1 C-BML schemas documented in this specification. In the latter case, compliant C-BML expressions shall validate (in the XML sense) against user-defined schemas that employ Phase 1 C-BML schema constructs documented in this specification, through use of the XML *include* or *import* statements. This latter use case permits widespread adoption of the standard while investigating the need for extension and enrichment of C-BML in future development phases.

With inclusion of “light” constructs in the Phase 1 C-BML schemas, additional levels of compliance to the Phase 1 C-BML specification are applicable. The specification enables construction of C-BML expressions using “light” data types and elements as described earlier or using the remaining data types and elements provided in the schemas. A system is ***fully compliant*** with either the “light” or remaining portion of the schemas when it is capable of *exchanging and using* all valid C-BML expressions that can be constructed from the respective portion of the Phase 1 C-BML schemas. A system is ***minimally compliant*** with either portion of the Phase 1 C-BML schemas when it is able to *exchange and use* all mandatory elements of valid C-BML expressions that can be constructed from the respective portion of the Phase 1 C-BML schemas.

8 Distribution Packaging of the Phase 1 C-BML Standard

The full set of Phase 1 C-BML products will be made available to the community in a zip archive format through the SISO Digital Library (http://www.sisostds.org/DigitalLibrary.aspx). The archive file contains the following folders and files:

- CBML_Phase1 (top-level folder)
    - Guidelines (folder) (**NOTE: This material is provided as an aid in understanding and applying the Phase 1 C-BML standard. The Guidelines are not yet approved through the SISO balloting process. The materials identified here are the draft materials provided at the time of balloting of the Phase 1 standard and do not reflect changes made in the Phase 1 C-BML Specification as a result of the product balloting and final approval. As the Guidelines and supporting materials complete development and balloting, they will be changed to fully conform to this Specification.**)
        - cbml-example-expressions (folder containing examples of the use of the Phase 1 XML schema structures; refer to the Guidelines document for details)
            - example-expressions (refer to the Phase 1 C-BML Guidelines document for descriptions)
                - ATO_ACO (folder)
                    - ACO_1.xml
                    - UAV_Mission_1.xml
                - CandidateTargetListType (folder)
                    - CandidateTargetListType_1.xml
                - EventWhatLocationType (folder)
                    - EventWhatLocationType_1.xml
                    - EventWhatLocationType_2.xml
                    - EventWhatLocationType_3.xml
                - EventWhatStatusType (folder)
                    - EventWhatStatusType_1.xml
                - HoldingTransferType (folder)
                    - HoldingTransferType_1.xml
                - NetworkServiceStatusType (folder)
                    - NetworkServiceStatusType_1.xml
                    - NetworkServiceStatusType_2.xml
                - Order (folder)
                    - Order_5.xml
                - OrganisationStructureType (folder)
                    - OrganisationStructureType_1.xml
                - Request (folder)
                    - BinaryAssociationRequest.xml
                    - BinaryHoldingRequest.xml
                    - BinaryItemPresenceRequest.xml

        - BinaryTypePresenceRequest.xml
        - BinaryTypeRequest.xml
        - Request_2.xml
        - UnitaryTypeRequest.xml
    - RequestAnswerReport (folder)
        - BinaryAssociationRequestAnswer.xml
        - BinaryHoldingRequestAnswer.xml
        - BinaryItemPresenceRequestAnswer.xml
        - BinaryTypePresenceRequestAnswer.xml
        - BinaryTypeRequestAnswer.xml
        - UnitaryTypeRequestAnswer.xml
    - RouteWhere (folder)
        - RouteWhere_CorridorArea.xml
        - RouteWhere_Line.xml
        - RouteWhere_Phaseline.xml
        - RouteWhere_Point.xml
        - RouteWhere_SurfaceVolumeCorridorArea.xml
        - RouteWhere_SurfaceVolumeCorridorArea.xml
    - SPOTReport (folder)
        - SPOTReport_1.xml
    - TargetPersonnelProtectionType (folder)
        - TargetPersonnelProtectionType_1.xml
    - TaskWhatStatusType (folder)
        - TaskWhatStatusType_1.xml
        - TaskWhatStatusType_2.xml
        - TaskWhatStatusType_3.xml
        - TaskWhatStatusType_6.xml
    - WhatEffectType (folder)
        - WhatEffectType_1.xml
    - WhoAddressType (folder)
        - WhoAddressType_1.xml
    - WhoAffiliationType (folder)
        - WhoAffiliationType_1.xml
    - WhoAssociationType (folder)
        - WhoAssociationType_1.xml
        - WhoAssociationType_2.xml
    - WhoHoldingType (folder)

- WhoHoldingType_1.xml
- WhoHoldingType_2.xml
- WhoHoldingType_3.xml

- WhoHostilityType (folder)
  - WhoHostilityType_1.xml
- WhoLocationType (folder)
  - WhoLocationType_1.xml
  - WhoLocationType_1a.xml
  - WhoLocationType_2.xml
  - WhoLocationType_3.xml
- WhoStatusType (folder)
  - WhoStatusType_1.xml
  - WhoStatusType_2.xml
- WhoTypeType (folder)
  - WhoTypeType_1.xml
  - WhoTypeType_2.xml
  - WhoTypeType_3.xml
  - WhoTypeType_4.xml
  - WhoTypeType_5.xml

- example-expressions-schema (folder)
  - cbml-example-expressions.xsd (XML schema for the example expressions provided in the Guidelines document, constructed from the normative schemas presented in this Specification document)
  - schema documentation (folder)
    - cbml-example-expressions-schema.html (Hyper-Text Markup Language file describing the structure and content of the cbml-example-expressions.xsd XML schema file)
    - cbml-example-expressions-schema_diagrams (folder; contains 21 files in Portable Network Graphics format supporting the HTML documentation of the XML schema file)

- document (folder)
  - SISO-GUIDE-002-2012-DRAFT.pdf (Phase 1 C-BML Guidelines document)
- supportingMaterials (folder)
  - images (folder)
    - C-BML Logical Layers.jpg (graphic of interrelationships across primary components of the C-BML logical data model)

- cbml-expressions.png (graphic of the top-level schema structure for C-BML expressions (order, request, and report))

- XMIRepresentation (folder)
  - C-BML Logical Layers.xml (XML Metadata Interchange (XMI) representation of the C-BML logical data model)
- CBMLtoJC3IEDM_Specification.xls (Microsoft Excel spreadsheet documenting a mapping from the Phase 1 C-BML XML schema structures to the JC3IEDM)
- C_BML-Light-Logical-Mappings_Light.csl (text file, expressed in the Condensed Scripting Language, describing a mapping from the Phase 1 C-BML "light" portion of the XML schemas to the JC3IEDM logical data model)

- Schemas (folder; refer to Section 7 for description of the Phase 1 C-BML XML schemas)
  - cbml-1.0 (folder)
    - cbml (folder)
      - cbml-action-types (folder)
        - 1.0 (folder)
          - cbml-action-types.xsd
          - cbml-action-types.doc (auto-generated documentation of the schema file)
      - cbml-affiliation-types (folder)
        - 1.0 (folder)
          - cbml-affiliation-types.xsd
          - cbml-affiliation-types.doc (auto-generated documentation of the schema file)
      - cbml-codes (folder)
        - 1.0 (folder)
          - cbml-codes-types.xsd
          - cbml-codes-types.doc (auto-generated documentation of the schema file)
      - cbml-composites (folder)
        - 1.0 (folder)
          - cbml-composites.xsd
          - cbml-composites.doc (auto-generated documentation of the schema file)
      - cbml-composites-light (folder)
        - 1.0 (folder)
          - cbml-composites-light.xsd
          - cbml-composites-light.doc (auto-generated documentation of the schema file)

- cbml-entity-types (folder)
  - 1.0 (folder)
    - cbml-entity-types.xsd
    - cbml-entity-types.doc (auto-generated documentation of the schema file)
- cbml-facility-types (folder)
  - 1.0 (folder)
    - cbml-facility-types.xsd
    - cbml-facility-types.doc (auto-generated documentation of the schema file)
- cbml-feature-types (folder)
  - 1.0 (folder)
    - cbml-feature-types.xsd
    - cbml-feature-types.doc (auto-generated documentation of the schema file)
- cbml-location-types (folder)
  - 1.0 (folder)
    - cbml-location-types.xsd
    - cbml-location-types.doc (auto-generated documentation of the schema file)
- cbml-materiel-types (folder)
  - 1.0 (folder)
    - cbml-materiel-types.xsd
    - cbml-materiel-types.doc (auto-generated documentation of the schema file)
- cbml-organisation-types (folder)
  - 1.0 (folder)
    - cbml-organisation-types.xsd
    - cbml-organisation-types.doc (auto-generated documentation of the schema file)
- cbml-person-types (folder)
  - 1.0 (folder)
    - cbml-person-types.xsd
    - cbml-person-types.doc (auto-generated documentation of the schema file)
- jc3iedm-codes (folder)
  - 3.0.2 (folder)
    - jc3iedm-codes.xsd

- jc3iedm-codes.doc (auto-generated documentation of the schema file)

- jc3iedm-simple-types (folder)
  - 3.0.2 (folder)
    - jc3iedm-simple-types.xsd
    - jc3iedm-simple-types.doc (auto-generated documentation of the schema file)

- Specification (folder)
  - SISO-STD-011-2013-DRAFT.pdf (the present document)

9 Summary

This Phase 1 C-BML Specification establishes the following standard format, content, and practices for the initial standardization of C-BML:

- JC3IEDM Edition 3.0.2 as the underlying data model to define a common set of terms and concepts for constructing C-BML expressions
- Established MIP practices for identifying and describing proposed changes to the JC3IEDM for C-BML applications
- Basic vocabulary and XML schemas describing the structure and content of the fundamental C-BML information components (the 5Ws) and an initial class of C-BML expressions

This specification provides basic building blocks for early adopters of the C-BML standard. Follow-on phases in C-BML standard development will (1) formalize the grammar for expressing plans, orders, requests, and reports, creating greater precision in the use of the C-BML 5Ws to encode particular doctrine (Phase 2), and (2) will formalize the semantics of plans, orders, requests, and reports to enable automated checking of the operational validity of the content of C-BML expressions (Phase 3).

Annex A Phase 1 C-BML XML Schema File: jc3iedm-codes.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/jc3iedm-codes/3.0.2/jc3iedm-codes.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/jc3iedm-codes/3.0.2/jc3iedm-codes.doc

Annex B Phase 1 C-BML XML Schema File: jc3iedm-simple-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/jc3iedm-simple-types/3.0.2/jc3iedm-simple-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/jc3iedm-simple-types/3.0.2/jc3iedm-simple-types.doc

Annex C Phase 1 C-BML XML Schema File: cbml-codes.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-codes/1.0/cbml-codes.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-codes/1.0/cbml-codes.doc

Annex D Phase 1 C-BML XML Schema File: cbml-entity-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-entity-types/1.0/cbml-entity-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-entity-types/1.0/cbml-entity-types.doc

Annex E Phase 1 C-BML XML Schema File: cbml-composites.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-composites/1.0/cbml-composites.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-composites/1.0/cbml-composites.doc

Annex F Phase 1 C-BML XML Schema File: cbml-composites-light.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-composites-light/1.0/cbml-composites-light.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-composites-light/1.0/cbml-composites-light.doc

Annex G Phase 1 C-BML XML Schema File: cbml-action-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-action-types/1.0/cbml-action-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-action-types/1.0/cbml-action-types.doc

Annex H Phase 1 C-BML XML Schema File: cbml-affiliation-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-affiliation-types/1.0/cbml-affiliation-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-affiliation-types/1.0/cbml-affiliation-types.doc

Annex I Phase 1 C-BML XML Schema File: cbml-facility-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-facility-types/1.0/cbml-facility-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-facility-types/1.0/cbml-facility-types.doc

Annex J Phase 1 C-BML XML Schema File: cbml-feature-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-feature-types/1.0/cbml-feature-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-feature-types/1.0/cbml-feature-types.doc

Annex K Phase 1 C-BML XML Schema File: cbml-location-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-location-types/1.0/cbml-location-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-location-types/1.0/cbml-location-types.doc

Annex L Phase 1 C-BML XML Schema File: cbml-materiel-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-materiel-types/1.0/cbml-materiel-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-materiel-types/1.0/cbml-materiel-types.doc

Annex M Phase 1 C-BML XML Schema File: cbml-organisation-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-organisation-types/1.0/cbml-organisation-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-organisation-types/1.0/cbml-organisation-types.doc

Annex N Phase 1 C-BML XML Schema File: cbml-person-types.xsd (Normative)

The schema file and associated documentation are provided in the Phase 1 C-BML distribution:

- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-person-types/1.0/cbml-person-types.xsd
- CBML_Phase1/Schemas/cbml-1.0/cbml/cbml-person-types/1.0/cbml-person-types.doc

Annex O Bibliography (Informative)

[B1] Blais, C. (2008). "Military Scenario Definition Language (MSDL): How Broadly Can It Be Applied?" Paper 08S-SIW-002. Proceedings of the Spring Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. April.

[B2] Borgers, E., Huiskamp, W., de Reus, N., and Voogd, J. (2007). "Research and Development towards Application of MSDL and C-BML in the Netherlands." Paper 07S-SIW-025. Proceedings of the Spring Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. March.

[B3] Bresnan, J. (ed.) (1982). The Mental Representation of Grammatical Relations. MIT Press. Cambridge, Massachusetts.

[B4] Bresnan, J. (1982). The Passive in Lexical Theory. In *The Mental Representation of Grammatical Relations*, J. Bresnan (ed.). MIT Press. Cambridge, Massachusetts.

[B5] Carey, S., M. Kleiner, M. R. Hieb, and R. Brown (2001). "Standardizing Battle Management Language – A Vital Move Towards the Army Transformation." *Proceedings of the Fall 2001 Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization. September.

[B6] Collins (2009). *Collins English Dictionary, 10th Edition*. William Collins, Sons, & Co.

[B7] Department of the Army (2008). Field Manual 3-0 – Operations. Available at: http://www.fas.org/irp/doddir/army/fm3-0.pdf.

[B8] Franceschini, R. W., D. J. Franceschini, R. B. Burch, R. Sherrett and J. Abbott (2004). "Specifying Scenarios Using the Military Scenario Definition Language." *Proceedings of the Fall 2004 Simulation Interoperability Workshop*, Simulation Interoperability Standards Organization. September.

[B9] Gruber, T. R. (1993). "A translation approach to portable ontologies." *Knowledge Acquisition*. 5:2, pp 199-220.

[B10] Gustavsson, P. M. (2011). Modelling, Formalising, and Implementing Intent in Command and Control Systems. Dissertation. De Montfort University. December.

[B11] Haarmann, B., and Schade, U. (2012). "Complementing Battle Management Language by Ontological Means." Paper 12S-SIW-006. Proceedings of the Spring Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. March.

[B12] Hieb, M. R., J. M. Pullen, W. P. Sudnikovich, and A. Tolk (2004). "Extensible Battle Management Language (XBML): A Methodology for Web Enabling Command and Control for Network Centric Warfare." *Proceedings of the Command and Control Research and Technology Symposium*. April.

[B13] Hieb, M. R., A. Tolk, W. P. Sudnikovich, and J. M. Pullen (2004). "Developing Extensible Battle Management Language to Enable Coalition Interoperability." *Proceedings of the 2004 European Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization. June.

[B14] Institute of Electrical and Electronics Engineers (IEEE) (1990). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York, NY.

[B15] Kaplan, R. M., and J. Bresnan (1994). Lexical-functional grammar: A formal system for grammatical representation. In *Formal Issues in Lexical Functional Grammar*. M. Dalrymple, R. M. Kaplan, and J. T. Maxwell III (ed.).

[B16] Levine, S., C. Blais, M. Hieb, J. Kearley, C. Pandolfo, M. Pullen, and J. Roberts (2007). "Joint Battle Management Language (JBML) – Phase 1 Development and Demonstration Results." *Proceedings of the Fall 2007 Simulation Interoperability Workshop*, Simulation Interoperability Standards Organization. September.

[B17] Merriam-Webster Dictionary (2013). Merriam-Webster, Incorporated.

[B18] Modeling and Simulation Coordination Office (2013). M&S Glossary. Version 2013.1. 1 July. Available at: http://www.msco.mil/MSGlossary.html

[B19] National Institute of Standards and Technology (NIST) (1993). Integration Definition for Information Modeling (IDEF1X). Federal Information Processing Standards Publication 184. National Institute of Standards and Technology. 21 December.

[B20] Norbraten, T. (2004). Utilization of forward error correction (FEC) techniques with extensible markup language (XML) schema-based binary compression (XSBC) technology. Master's Thesis. Naval Postgraduate School. Monterey, California. December.

[B21] North Atlantic Treaty Organization (NATO) (2000). Standardisation Agreement (STANAG) 2014. Formats for Orders and Designation of Timings, Locations, and Boundaries. 17 October.

[B22] North Atlantic Treaty Organization (NATO) (2010). AJP-01 Allied Joint Doctrine. Brussels, Belgium. December. Available at http://nsa.nato.int/nsa/zPublic/ap/AJP-01(D).pdf.

[B23] North Atlantic Treaty Organization (NATO) (2013). NATO Glossary of Terms and Definitions. AAP-06 Edition 2013. North Atlantic Treaty Organization Standardization Agency.

[B24] Pullen, J. M. (2005). "Using Web Services to Integrate Heterogeneous Simulations in a Grid Environment." *Journal on Future Generation Computer Systems.* Volume 21. pp. 97-106.

[B25] Pullen, J. M., Alstad, A., Brook, A., Corner, D., Mevassvik, O. M., and Wittman, R. (2012). "MSDL and C-BML Working Together for NATO MSG-085." Paper 12S-SIW-045. Proceedings of the Fall Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. March.

[B26] Remmersmann, T., El Abdouni Khayari, R., Gautreau, B., Khimeche, L., and Schade, U. (2012). "Lessons Recognized: How to Combine BML and MSDL." Paper 12S-SIW-012. Proceedings of the Fall Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. March.

[B27] Riggs, W. C. (2011). "The Representation of Operational Terms and Graphics in Simulation Standards: Emerging Results and Issues." Paper 11F-SIW-057, *Proceedings of the Fall Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization. September.

B[28] Schade, U., and M. R. Hieb (2006). Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests, and Reports. In *11th International Command and Control Research and Technology Symposium*. Cambridge, UK: DoD Command and Control Research Program. June. Available at: http://www.dodccrp.org/events/11th_ICRRTS/html/papers/069.pdf

[B29] Schade, U., and M. R. Hieb (2007). Improving Planning and Replanning: Using a Formal Grammar to Automate Processing of Command and Control Information for Decision Support. *C2 Journal*. Vol 1, No 2. June. Available at: http://www.dodccrp.org/files/IC2J_v1n2_04_Schade.pdf

[B30] Schade, U., and Hieb, M. R. (2007). "Battle Management Language: A Grammar for Specifying Reports." Paper 07S-SIW-036. Proceedings of the Spring Simulation Interoperability Workshop. Simulation Interoperability Standards Organization. March.

[B31] Schade, U., Hieb, M. R., Frey, M., and Rein, K. (2010). Command and Control Lexical Grammar (C2LG) Specification. Technischer Bericht ITF/2010/02. Fraunhofer-Institut fur Kommunikation, Informationsverabeitung und Ergonomie FKIE. July.

[B32] Sudnikovich, W., M. R. Hieb, M. Kleiner, and R. Brown (2004). "Developing the Army's Battle Management Language Prototype Environment." *Proceedings of the 2004 Spring Simulation Interoperability Workshop*. Simulaiton Interoperability Standards Organization. April.

[B33] Tolk, A., and C. Blais (2005). "Taxonomies, Ontologies, and Battle Management Languages – Recommendations for the C-BML Study Group." *Proceedings of the Spring 2005 Simulation Interoperability Workshop*. Simulation Interoperability Standards Organization. April.