18TH ICCRTS

Eighteenth International Command and Control Research and Technology Symposium

C2 IN UNDERDEVELOPED, DEGRADED AND DENIED OPERATIONAL ENVIRONMENTS

June 19 - 21, 2013

Alexandria, Virginia USA

Topics: Experimentation, Metrics, and Analysis
Modelling and Simulation
Architectures, Technologies and Tools

Title: Implementing a Standards Development Framework for the Coalition Battle Management Language

Authors: **Kevin Heffner (POC)**
Pegasus Research and Technologies
PO Box 47552
CP Plateau Mont-Royal
Montreal (QC) H2H 2S8 Canada
+1-514-360-4920
k.heffner@peretec.com

Kevin Gupton
Applied Research Laboratories
The University of Texas at Austin
10,000 Burnet Rd.
Austin, Texas 78758
512-835-3679
kgupton@arlut.utexas.edu

Implementing a Standards Development Framework for the Coalition Battle Management Language

Kevin Heffner, Kevin Gupton

Abstract

The Coalition Battle Management Language (C-*BML) is an open standard being developed for the exchange of digitized military information among command and control, simulation and autonomous systems by the Simulation Interoperability Standards Organization (SISO) and is no exception in this regard. As the first phase of the C-BML standard nears release, the Phase 2 Drafting Group (DG) has proposed a framework to identify and track the concerns and requirements to be addressed in the next major C-BML standard release. The C-BML Standard Development Framework (SDF) proposed in 2012 organizes various parts of C-BML and establishes a separation-of-concern in terms of function and scope. This paper reports on the preliminary results of implementing the C-BML SDF. In particular, the reference architecture that is the basis for the framework is described as well as experience and insight gained in the handling of stakeholder requirements. Also discussed are lessons learned and challenges faced in the effective use of applied ontology as an integral part of the standard development.*

1. Introduction

SISO currently is developing C-BML, a standardized formal language for the exchange of digitized military information among command and control, simulation and autonomous systems. C-BML is an interoperability standard that can facilitate greatly the preparation and execution of military scenarios in support of military enterprise activities such as: *Training*; *Support to Operations*; and *Concept Development and Experimentation*. Preliminary research using C-BML already has shown the benefits that can be achieved: 1) reduced exercise/experiment/planning preparation time; 2) increased realism of the training/experimentation environment; and 3) reduced cost associated with the decrease in the number of required simulator operators.

The idea for a Battle Management Language to standardize the exchange of information among Command and Control (C2) and simulation systems was introduced as early as 1999 by Argo et al. by the US Army [1]. In 2004, SISO decided to form a study group to consider the need for a BML at the coalition level or a *Coalition* BML or C-BML. The study group findings and recommendations lead to the SISO C-BML Product Nomination and the formation of the C-BML Product Development Group (PDG) [2] to develop C-BML as an open, international standard.

Consistent with the C-BML Study Group recommendations, the C-BML PDG defined a three phase product development plan. This plan called for three overlapping phases that would define: 1) a C-BML vocabulary; 2) a C-BML grammar; and 3) a C-BML ontology.

1.1. C-BML Phase 1

The Coalition Battle Management Language (C-BML) development activity recently has reached a major milestone, with the successful balloting of the C-BML Phase 1 standard product. The development of C-BML Phase 1 has been plagued by many difficulties and challenges [3] and it has taken more than six years to produce the first in a series of three versions comprising the C-BML standard. Indeed, developing complex technical interoperability standards such as C-BML, that involve a diverse set of stakeholders has proven difficult [3], and this has been the case in other domains as well [4]. Reference [4] proposes a systems engineering approach to building and maintaining technical interoperability standards, similar to the approach advocated by Lang et al for the Multilateral Interoperability Programme (MIP) Block 4 Working Group [5].

The initial C-BML Product Nomination specifies the interdependence of C-BML and the SISO Military Scenario Definition Language (MSDL), intended for simulation initialization, with the general understanding was that MSDL and C-BML should be merged or at least deconflicted at some point.

This has become an important aspect of the C-BML standard development and will be discussed below.

1.2. C-BML Phase 2

Gupton and Heffner [6] addressed the problems faced by the SISO C-BML Phase 1 development activity and proposed a Standards Development Framework (SDF) to facilitate the C-BML Phase 2 standard development effort. Beyond the C-BML Phase 2 product, one of the underlying goals of developing the SDF was to propose a means to evolve the C-BML standard based on a set of traceable requirements in a controlled, repeatable manner while allowing for community extensions without sacrificing interoperability. The SDF organizes the various aspects of C-BML that may be considered in the next C-BML version, establishing a separation-of-concern at different levels of abstraction in terms of function and scope. Described in this paper are the delineating layers of a reference architecture that is the basis for the framework and how it relates to implementation-specific and technology-specific issues. The framework aims to help organize the C-BML development effort, adding clarity to scope discussions and facilitating future standard product development. The paper also presents some of the early results of implementing the SDF in conjunction with the MIP Information Model (MIM) and describes some of the practical aspects of how the SDF can be used as a means to align an emerging standard with other existing standards and specifications.

Whereas C-BML Phase 1 primarily has focused on establishing a controlled vocabulary while Phase 2 focus areas include the C-BML grammar, message metadata, transport, and the definition of C-BML services. Over the course of developing the C-BML Phase 1 draft product, many lessons were learned, prototypes were developed, requirements were refined, and expectations have evolved.

With the C-BML PDG endorsement, the C-BML Phase 2 development activity commenced in 2011. The Phase 2 C-BML Drafting Group (DG) is building upon the C-BML Phase 1 product while addressing three main areas of concern: Requirements, Reference Architecture, and Implementation.

1.3. C-BML Standard Development Framework (SDF)

The C-BML Standards Development Framework (SDF) has been developed to support the following activities:

1) C-BML Requirements Management;
2) Development of C-BML conceptual, logical and physical model representations;
3) C-BML grammar development;

4) Definition of message transport and C-BML services;
5) Creation a set of examples that clearly illustrate the use of C-BML; and
6) Communication of the various C-BML products and related documents to stakeholders.

The C-BML SDF establishes architectural continuity: from requirements to reference architecture, to implementation-specific and technology-specific architecture; and finally to application-specific architecture and reference implementations. One of the intended uses of the standard is to have C-BML-compliant systems that can be integrated into various operational environments. Therefore, C-BML SDF provides an extension mechanism for coalition, national, service, and organization-specific applications. Consistent with the MoDAF, DoDAF, and NAF architectural frameworks, the C-BML SDF allows for describing how use-cases for operational mission threads and message threads can be supported as part of a C-BML-based solution.

1.4. C-BML Examples

C-BML is intended to be an unambiguous, formal, language for communicating military information for machine-to-machine communication. In general terms, a *grammar* is a set of rules that dictate what valid sentences or expressions can be constructed for a given language.

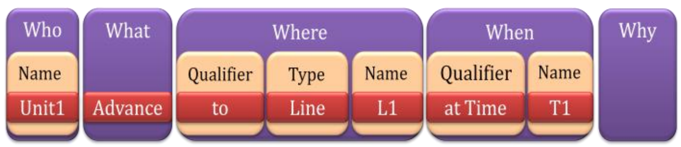

Figure 1 Graphical C-BML Example illustrating 5Ws

Argo et al. suggested that the BML expressions be based on a structure that included 5Ws to facilitate the programming of simulated/automated units: Who What Where When Why [1]. The 5Ws can be described as follows:

- **Who**: The tasking unit; The tasked unit; The supported unit; The supporting unit; The target; The reporting unit; The object of a report.
- **What**: The type of operation or task to be executed; The event being observed
- **Where**: Where is the task to be executed; Where is the event being observed
- **When**: The time the task to be executed or has been executed; the time an event observed.
- **Why**: The purpose, motivation, desired effect or result.

C-BML has followed these basic definitions. A graphical example of a simple C-BML task is shown in Figure 1, (the Why has not been included for clarity).

In practice, C-BML expressions will be communicated using one of several concrete syntaxes such as the eXtensible Markup Language (XML) or Java Serialized Object Notation (JSON). An example of a simplified

```
<Order>
    <AirTask>
        <TaskeeWho>
            <UnitID>CA-UAV</UnitID>
        </TaskeeWho>
        <What>
            <ActionCode>AI</ActionCode>
        </What>
        <Where>
            <WhereID>14010000784100000427</WhereID>
            <WhereLocation>
                <GDC>
                    <Lat>40.062195</Lat>
                    <Lon>47.57694</Lon>
                    <ElevAGL>3000.0</ElevAGL>
                </GDC>
            </WhereLocation>
        </Where>
        <StartWhenTime>
            <StartTimeQualifier>AT</StartTimeQualifier>
            <DateTime>20091022141229.359</DateTime>
        </StartWhenTime>
        <AffectedWho>
            <UnitID>OMF195-B12</UnitID>
        </AffectedWho>
        <TaskID>14099999000000000019</TaskID>
    </AirTask>
    <OrderIssuedTime>20091022141443.000</OrderIssuedTime>
    <OrderID>14099999000000000030</OrderID>
    <TaskerWho>
        <UnitID> 1-HBCT</UnitID>
    </TaskerWho>
</Order>
```

Figure 2 Simplified XML C-BML example

XML expression for an Air Interdiction task is shown in Figure 2.

This paper first addresses why a C-BML SDF is needed. Then the C-BML SDF is described in terms of its layers and components. Finally, the benefits and the impact that the C-BML SDF can have on the standardization activity are considered, including preliminary results on implementing the SDF.

2. Motivation for a C-BML Standards Development Framework

C-BML is being developed by SISO as a set of specifications to facilitate the standardized exchange of military information such as orders, plans, reports, and requests among Command and Control (C2), simulation and autonomous systems [2].

C-BML can be characterized or described in several ways, but essentially C-BML provides a common, standardized interface with a vocabulary and grammar of sufficient expressiveness to support reporting and tasking among real, simulated, or robotic forces. C-BML expressions are intended to be unambiguous[1] and parsable and therefore C-BML ultimately will define a formal language in a mathematical representation that allows for automated processing. Technology-independent and protocol agnostic, the C-BML standard also specifies information required to proceed with the actual transfer and exchange of C-BML messages using different transport mechanisms via C-BML services.

2.1. Defining the Scope of C-BML

The C-BML standard is intended to cover multiple domains (Air, Land, and Maritime), multiple echelons (from dismounted soldier to Brigade/Division) for multiple nations. One of the biggest challenges in drafting the C-BML standard is that of defining the scope and requirements for C-BML. Reference [15] defines a starting point for scoping C-BML, but the lack of a validated set of stakeholder requirements for C-BML has been a key issue throughout the standard development.

For example, Air Tasking Orders (ATO), Land Forces Operations Orders (OPORD) and Maritime Forces Operation General Matter (OPGEN) typically are large documents with much information captured as free-text and/or in annexes. If the primary purpose of C-BML is to communicate such orders from C2 systems for execution by simulated forces in simulation systems, then consideration must be made for the subset of information that is required by the target simulation. It also is important to establish what information that is contained in free text is required or will be required by simulations. For example, Rules of Engagement (ROE) and Commander's Intent (CI) generally are expressed as free-text. But are ROE and CI required inputs into current simulation systems? If they are, then a translation mechanism may be required. The answer is not clear, but without a set of validated stakeholder requirements, there is little hope of clearly defining the scope of C-BML. Heffner [8] points out also that it is important to distinguish between sustaining and disruptive changes when establishing requirements for new standards.

Similarly, hundreds of C4I system types, tactical messages, message threads, and mission threads exist today in support of various echelon levels. It is unrealistic to expect that C-BML might support them all completely in a practical timeframe.

One of the main challenges in the C-BML Phase 2 development activity is that C-BML stakeholders require support for force structures that include coalition, national, joint, and service-specific operations. Support also is required for several functional areas such as fire-support, logistics, intelligence, and others. C-BML stakeholders also include materiel solution developers and integrators who intend to implement and deploy C-BML-based technologies in heterogeneous environments with different configurations of simulations and C4I systems characterized by:

---

[1]Unambiguous refers here to the mathematical definition of formal grammars and implies the *existence of a unique derivation tree for a given expression*. Ambiguity in the *interpretation* of "well-formed" unambiguous C-BML expressions is addressed, in part, by the C-BML ontology.

- Presence of both legacy and new systems;
- Mix of old and new software technologies; and
- Varying levels of resolution, from single entities to full theatres of operation.

2.2. Establishing a C-BML Logical Data Model

The Phase 1 product establishes a controlled vocabulary in the form of a set of XML Schema Description (XSD) documents and is based on a set of basic elements taken from the MIP Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM), consistent with the recommendations in reference [2]. However, this schema has been hand-crafted and has grown quite complex and therefore it is difficult to apply changes or reuse model elements. An XML Schema Description (XSD) can be considered as a *model*, but in Model-Driven Architecture (MDA) terms [7], it is a Platform Specific Model (PSM). The MDA approach recommends building Platform Independent Models (PIM) sometimes referred to as Logical Data Models (LDM) and then generating PSM through well-defined and controlled model transformation. Applying model changes to a PIM generally is more practical than applying model changes directly in a PSM for all but the simplest of models. Therefore, the SDK seeks to establish a C-BML PIM or LDM as a Unified Modeling Language (UML) model and to generate one or more PSM automatically using UML transforms. There are several immediate advantages to this approach. First of all, one no longer is limited to XSD as the only PSM. Certainly XML has become one of the *de facto* standards for data exchange, but JAVA Serialized Objects (JSON), for example, increasingly is being utilized in conjunction with the Representational State Transfer or RESTful services approach. Another advantage is that, arguably, it is easier to maintain and evolve a UML model than it is to maintain and evolve a set of XSD schemas.

2.3. Operational Tactical Messages and C-BML

The C-BML standard aims to standardize exchange of digitized military information among C2, simulation and autonomous systems, primarily at the tactical level. In the past, and in current operations, much of this information has been—and continues to be—communicated using voice comms, chat, and formatted tactical message sets, such as the NATO Allied Procedural Publication APP-11(C) [9] and US Army FM-6-99 [10].

These communication means are at the core of modern military operations and it is not realistic to consider replacing them with fully automated systems in the short-term. However, as modern military forces undergo transformations and introduce capabilities that leverage new technologies into the military enterprise, it is recognized that these message sets have inherent limitations regarding the extent to which they can be utilized as part of automated information flows [11]. Nonetheless, they still reflect the operations procedures still in use and generally are consistent with the way that the armed forces for whom they were defined conduct business today.

But as machines become more intelligent and agent and automation technologies continue to make their way into modern C2, simulation and autonomous systems, existing tactical messages sets will need to evolve, and in some instances likely make way for formal language message sets that are better suited for use in automated business processes of the military enterprise. Over a decade ago, a significant milestone was the recognition of XML as a more suitable means for representing military Formatted Text Messages (FTM) as compared to the previous “teletype” format [12]. More recently, research efforts have been directed at considering interoperability issues beyond the message format, but that identify issues and problems due to the use of free-text and underlying differences in the data dictionaries and data models across user groups such as a joint or multinational force [13][14].

Therefore, the use of traditional FTM tactical message sets as the basis for interoperability in new capabilities can be problematic and must be done with great care to avoid situations such as semantic misalignment [13]. However, in spite of the shortcomings of these FTM tactical message sets and even if it is likely that future C2 systems will send and receive information automatically using other improved formats, the currently available FTM tactical message sets still are the basis for defining the information flows that support mission threads and other military use-cases.

Consequently, there is merit in the approach that considers information flows based on existing FTM tactical message sets in the context of specific operational mission threads that then can serve as a source of requirements for the C-BML standard. More specifically, these requirements can help to shape the C-BML model in terms of vocabulary, message metadata, transport metadata and information exchange considerations.

On the other hand, this approach imposes the need for a dedicated framework to manage the relationship between the tactical message sets, the related operational requirements, and the C-BML model.

2.4. Demystifying the C-BML Standard

Finally, the barriers to adoption for C-BML must be mitigated. The C-BML standard must serve a clear purpose, must be easy to understand and explain, must be extensible and flexible, and must integrate well into the mixed legacy solutions of operational environments. Better articulation of C-BML purpose, scope, applications, and role among other standards serves to improve discovery, evaluation, adoption, and sustainment of C-BML. Assessments that C-BML is too complicated must be addressed.

The C-BML SDF is intended to organize all of these concerns into a framework that facilitates discussion, resolution of conflicting requirements or interpretations, and aids in properly scoping and organizing the standardization activity.

3. C-BML Standard Development Framework

3.1. Key Features of the SDF

The C-BML SDF[2] was designed to have certain key features or quality attributes. Other valuable features emerged from creation of the SDF itself.

1) **Requirements-driven:** C-BML stakeholders have promoted many, sometimes seemingly contradictory requirements for a C-BML standard. C-BML must be generic enough to support multiple applications, yet extensible enough to be implementable in diverse environments. To achieve this, the framework must capture the various stakeholder requirements and allow for development, tracking, validation and other requirements management activities.
2) **Extensible:** Because of the breadth of stakeholder requirements, C-BML must strike a balance between generic utility and a specific solution. Where C-BML stops, implementers must be able to pick up the standard and make community-specific extensions to suit their domain. Thus, the SDF uses extensible design patterns to define a C-BML "core" and provide guidance for creating community extensions. Of course, those extensions could be later considered for inclusion in future versions of the C-BML standard.
3) **Maintainable:** C-BML will support a community where many legacy standards, systems, and networks must be integrated—and will inevitably evolve. To enable an environment of past, present, and future technologies, the SDF uses a layered approach to describe the conceptual, implementation-specific, and technology-specific aspects separately. The result is a solution that is both durable to changes in technology, will also supporting specific, implementable detailed designs.
4) **Phased development:** The broad requirements of C-BML stakeholders also sets the stage for a product that—if not carefully scoped—will "never be finished." To address this issue, the SDF endorses a phased, use case-driven, and capability-driven approach for development. The SDF uses a full spectrum of architectural views to define a product that is implementable and useful, but recommends a gradual development of a few capabilities at a time, thus better managing scope of C-BML development.

[2] The C-BML SDF is based in part on the US Joint Intelligence Community/DoD Content Discovery and Retrieval Model.

5) **Establishes artifacts needed to align with other standards:** MSDL, entity enumerations (SISO EWG), HLA, DIS, RPR, ANDEM, HSCB, BOM, and other specifications have some overlapping implementation or design concerns with C-BML concepts and technologies. Commonalities can be isolated to specific components of the SDF. For example, the need for semantic alignment is addressed by the *C-BML Content Model*, and the need for compatible XML schemas is addressed by SDF section addressing *Information Exchange Mechanisms*. This way of separating concerns adds clarity to what aspects of standards or systems need to be aligned with C-BML.

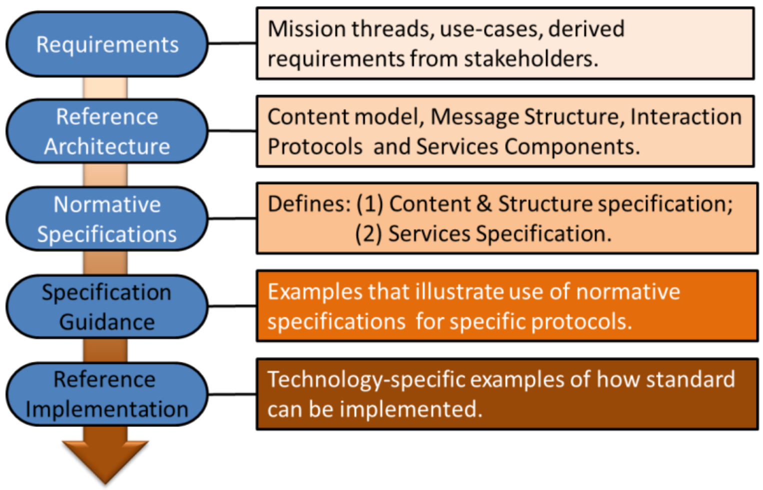

Figure 3 C-BML SDF Overview

3.2. C-BML SDF Overview

The C-BML SDF is defined as five components layers. The five layers separate levels of conceptualization and specification in the objective C-BML product. Figure 3 depicts the five-layer "stack," which is referenced throughout this paper as each layer is defined.

The five sections of the SDF are:

- **Requirements – Operational use cases defined in terms of mission threads, operational activities, and information flow.**
- **Reference Architecture** – Conceptual-level organization of views to support domain data models, message framework, interaction protocols, and abstract service components. This content is populated incrementally based on requirements.
- **Normative Specifications –** The implementation-specific details that form the C-BML specification. This specification relies on the reference architecture for structure, rationale, and integrity across multiple implementation options.
- **Specification Guidance** – The informative, non-binding guiding addendum to the normative specification. This may include technology-specific recommendations, example implementations, example extensions of the standard and other rationale not mandated as part of the normative specifications.
- **Reference Implementation (RI)** – Software implementations developed as part of the C-BML drafting process to validate the specifications and demonstrate it is implementable. This may actually be out of scope for a SISO PDG or DG, but it is included here for completeness. RIs might be developed through collaborations of C-BML stakeholders.

The next sections describe each of the SDF layers in more detail. As already described, each section builds upon the previous section, adding additional detail leading up to consistent, implementable, and interoperable specifications.

3.3. Requirements

The requirements layer of the SDF advocates the creation and selection of operational use cases to drive C-BML development. Each use case may be generally or specifically useful to all or part of the C-BML community, and each use case should exercise some capability that C-BML must support. Ideally, operational use cases will be defined uniformly in terms of mission threads, operational activities, and essential information flow definitions, as depicted in Figure 4.

Technical use cases and requirements are also key to the development of C-BML. The SDF does not prescribe how technical requirements must be captured, but it specifies that the Requirements layer is the appropriate area to do so and that the requirements should be linked to the C-BML products. However, the SDF requirements approach is consistent with the DoDAF, MoDAF and NAF architectural frameworks.

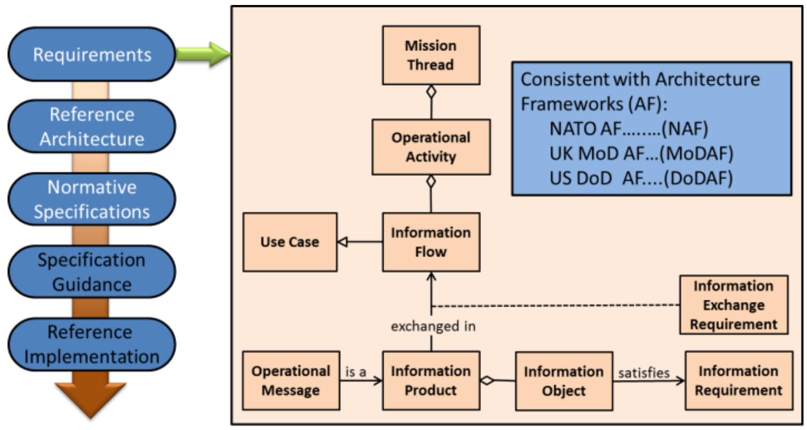

Figure 4: Requirements Model

3.4. Reference Architecture

The Reference Architecture (RA) layer of the SDF defines conceptual and abstract artifacts that describe what C-BML is and how it works. See reference [16] for a definition of Reference Architecture. This layer is arguably the most important section of the SDF, tying together all of the main elements of the standard into a set of viewpoints.

Much of the value that SDF is intended to provide is in the sections that comprise the RA. As shown in Figure 5 the RA defines a core model for data, messages, interchange protocols, and services—each of which is extendable by implementers. In addition to the core models themselves, the RA defines *how to specify* the models and the extensions. By defining the patterns or metamodel for C-BML components, the RA further enables architects and implementers to use C-BML beyond the parts provided by the specification alone.

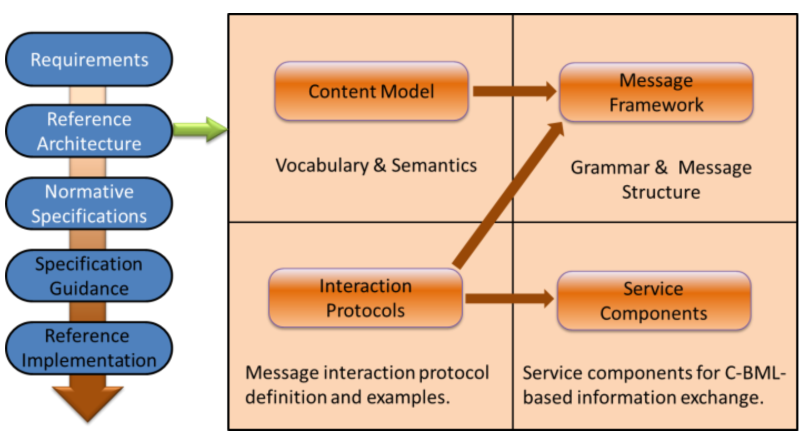

Figure 5 Reference architecture

Following the relationships shown in Figure 5, the next subsections describe each component of the reference architecture and the function it serves.

3.4.1. Reference Architecture: Content Model

The first component of the RA is the Content Model, shown in Figure 6. The Content Model is comprised of a Core Content Model and guidelines for extending the core for coalition, national, service, or system-specific applications.

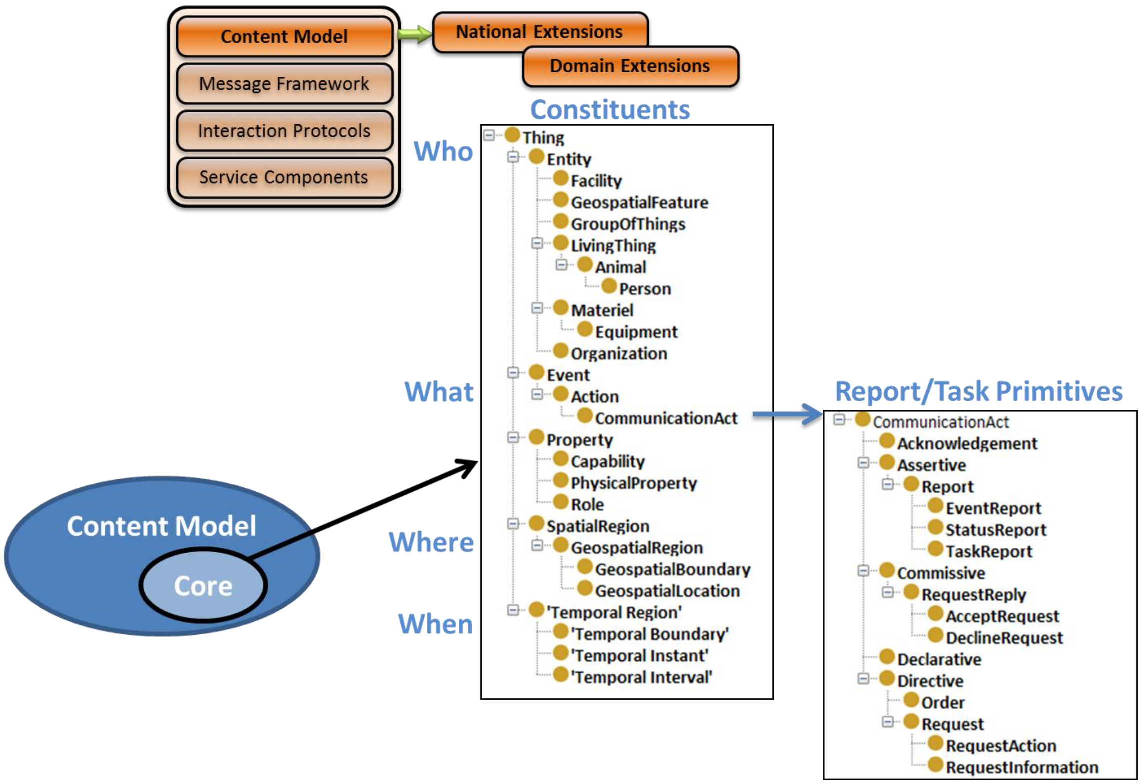

Figure 6 Content model; core and extensions

The Core Content Model is a domain ontology defined in the Web Ontology Language (OWL), but primarily managed in UML to take advantage of code generation utilities. By utilizing OWL, the Content Model benefits from the extensibility features described in [13]. The initial C-BML Product Development Plan called for the definition of an ontology product as the third of a three-phase approach. However, due to the maturity in the field of applied ontology and the strong tools support in this area, the definition of a C-BML ontology now has become a central part of the Phase 2 development activity. Therefore, consideration should be given to a possible redefinition of the three phases and the possible merging of Phase 2 and Phase 3. Indeed, the initial C-BML ontology defined in Phase 2, in all likelihood will fulfill the initial stakeholder expectations as expressed in the product nomination.

The Core Content Model is limited to those elements that are most essential to C-BML and are organized into three groups:

- General objects conventionally referred to in C-BML as the “Five W’s”. The Core Content Model generalizes the concepts to common ontological elements: Entity for “Who”; Event and Action for “What”; Spatial Region for “Where”; Temporal Region for “When”; “Why” has yet to be addressed [17].
- Properties, for describing state, status, capability, relationships, or any aspect of the general objects.

- Communicative acts, which are an essential part of communication theory. Their use has been advocated for by [17], among others. Communicative acts are types of Actions and include assertives (reports), commissives (replies), declaratives (declarations of control measures or task organization), and directives (orders and requests). Speech act theory accounts for other communicative acts [18], but these are not applicable to C-BML.
- Beyond vocabulary, the communicative acts are a central part of Phase 2 and previously were advocated by [17] and [19] in relation to the C-BML grammar. This part of the Core Content Model borrows from the IEEE Foundation for Physical Intelligent Agents (FIPA) standard [20] and research efforts to unify a Communication Ontology [21].

3.4.2. Reference Architecture: Message Framework

The Message Framework defines an abstract message structure that logically distinguishes the elements that comprise a C-BML message: *message content*, *message metadata*, and *transport metadata*, as shown in figure 7. At the implementation level, the transport metadata typically will be specified as a header that is part of an implementation-specific transport envelope, which might enclose the rest of the C-BML message (i.e. content and metadata). The Message Framework also defines how the information supported by the Content Model can be assembled into expressions and messages consistent with the *production rules* (i.e. the C-BML grammar).

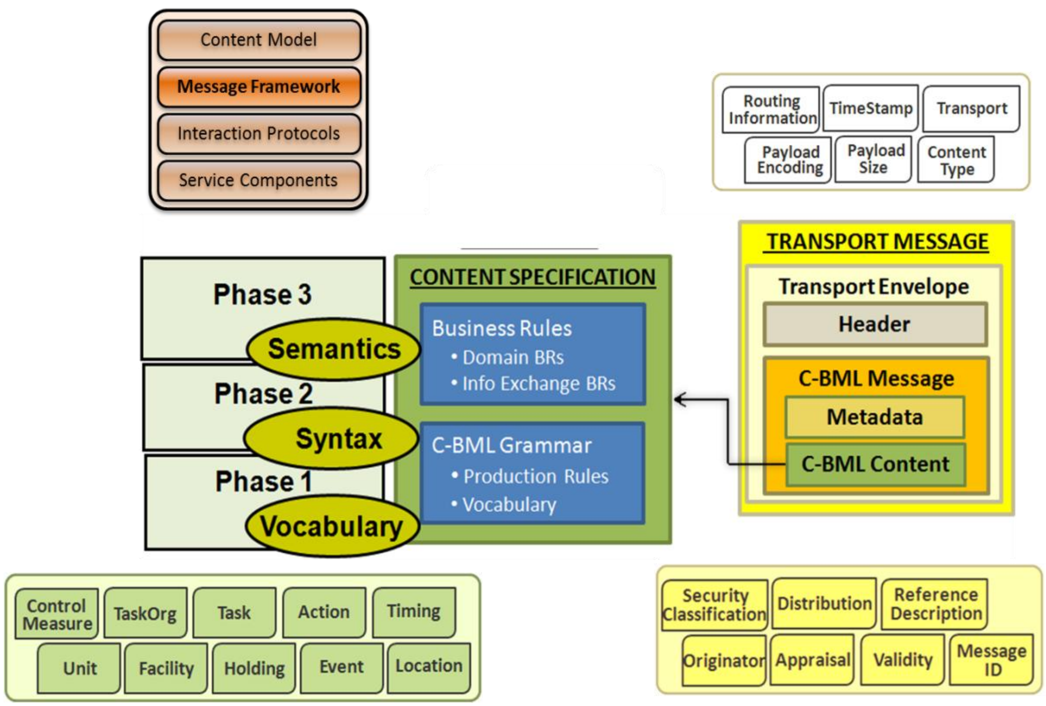

Figure 7 Message framework

The message framework specifies a set of high-level structures for constructing messages based on as set of rules and guidance, thus providing a consistent and standardized approach for generating C-BML messages. At the same time, it has the flexibility to allow users to build expressions and messages required for their community purposes.

3.4.3. Reference Architecture: Interaction Protocols

The Interaction Protocols section of the SDF defines how series of related message exchanges or "conversations" can be constrained to a protocol. Operation information exchanges are rarely limited to singular, independent messages. More often, multiple messages are exchanged among multiple actors with dependencies across messages. For example, an *acknowledgment* message may follow a *report* or *request*. Message exchanges that relate to a common mission context are referred to as "message threads." The Interaction Protocol section provides the constructs necessary to define protocols for C-BML messages, thus driving additional requirements to both message content, message metadata and system implementations.

As mentioned above, many aspects of tactical information exchange cannot be standardized in C-BML because of diversity of domains, organizations, echelons, and applications. For this reason, the SDF does not specify a set of *standard* interaction protocols, but rather defines how to capture interaction protocols in a structured form. Implementers then can catalog the interaction protocols, compose the protocols, determine which protocols are compatible, or not and eventually implement autonomous agents (robotics and simulations) to converse intelligently.

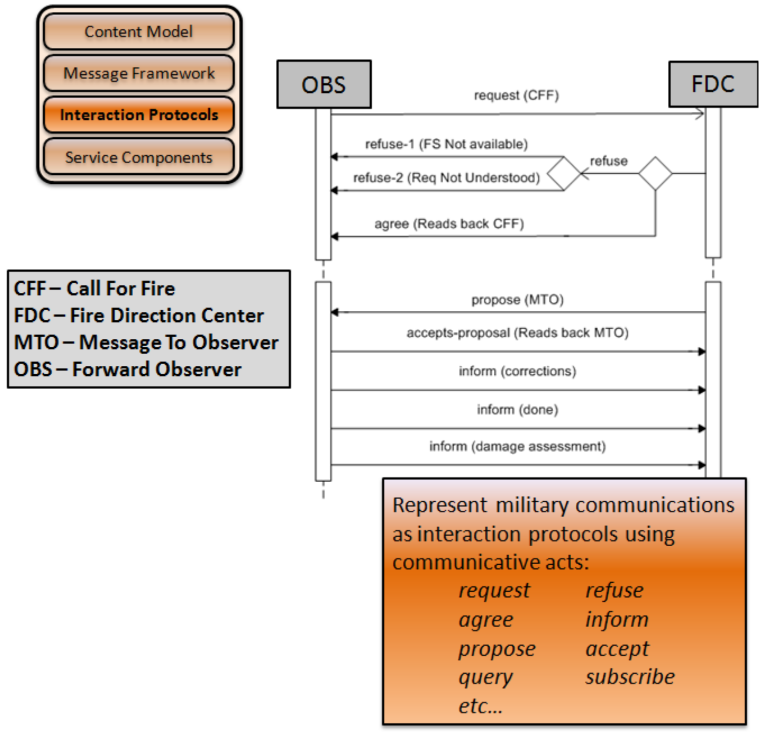

Figure 8 Example interaction protocol

Figure 8 depicts a notional Call-For-Fire interaction protocol – an example use case being developed to illustrate the SDF and one of several mission types that are the current focus of the C-BML Phase 2 development. This message thread occurs between the *Forward Observer* and the *Fire Detection Center* and involve a series of messages that each specify a communicative act, such as: *request*, *refuse*, *agree*, *propose*, *accepts-proposal*, *inform* etc.

3.4.4. Reference Architecture: Service Components

The Service Components section, currently under development, is intended to organize how service interfaces are to be defined in C-BML. This section defines the set of services (see Figure 9) that are made available through service interfaces that will be accessed or provided by C-BML systems in order to provide some capability. The core C-BML Services can be combined with other services to provide domain-specific or application-specific services. For instance, following the Call for Fire example in the previous section, an implementer might define services for Forward Observer (OBS), Fire Direction Center (FDC), or Fires Unit agents.

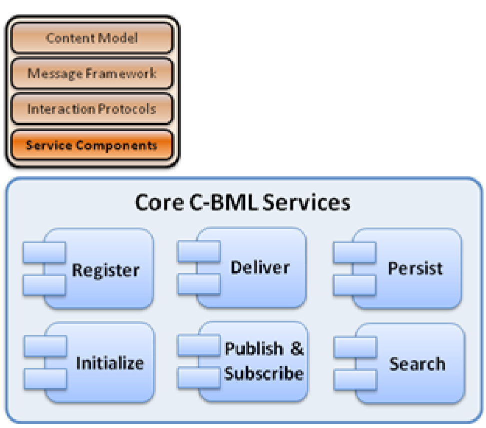

Figure 9 C-BML Services Components

In line with the same scope and extension approach outlined in the previous SDF sections, the Service Components section will define a limited set of service definitions for the core C-BML services, but primarily will guide communities in defining their own services in a common way so they may be cataloged, reused, compared, combined, etc. Service endpoints such as registration, producers, consumers, discovery, publishers, and subscribers need to be aligned to the enterprise service solutions of industry.

3.5. Normative C-BML Specifications

The Reference Architecture described in previous sections provides an overarching framework for organizing various aspects of C-BML. In the end, the C-BML specification must define a physically *implementable* product. The Normative Specification layer of the SDF combines elements from the Reference Architecture in a formal specification that then can be applied to the definition of a specific implementation.

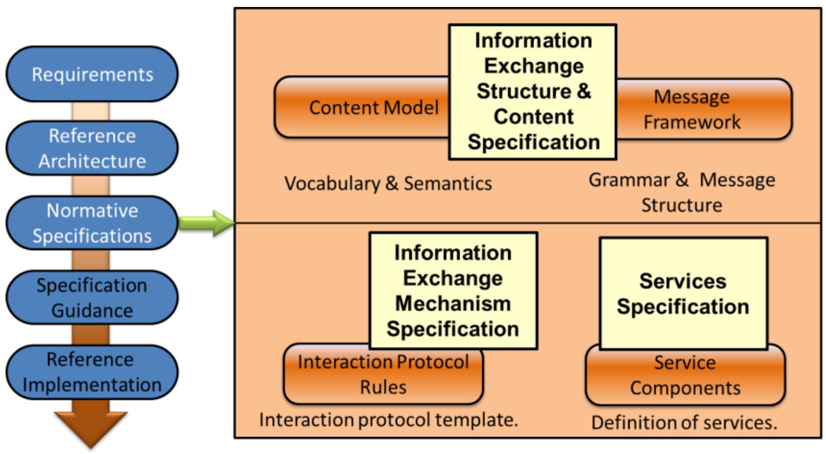

Figure 10 Normative Specifications

Implementable means the normative specifications allow for the application of the Reference Architecture's views to specific information exchange standards such as SOAP/WSDL and RESTful web services and XML Schema toward the goal of achieving some level of interoperability among systems.

Depicted in figure 10, the normative specifications will define a C-BML Content Model (vocabulary and ontology) and a Message Framework (grammar and transport). It also will specify how the Content Model relates to the Message Framework and the XML schemata. The normative specifications also will specify rules and templates for implementing Interaction Protocols and Service Components.

As part of the normative specifications, expressions and messages defined in the Message Framework are aligned semantically to the Content Model that begins to define how C-BML messages are to be interpreted by consumers, depicted in figure 11.

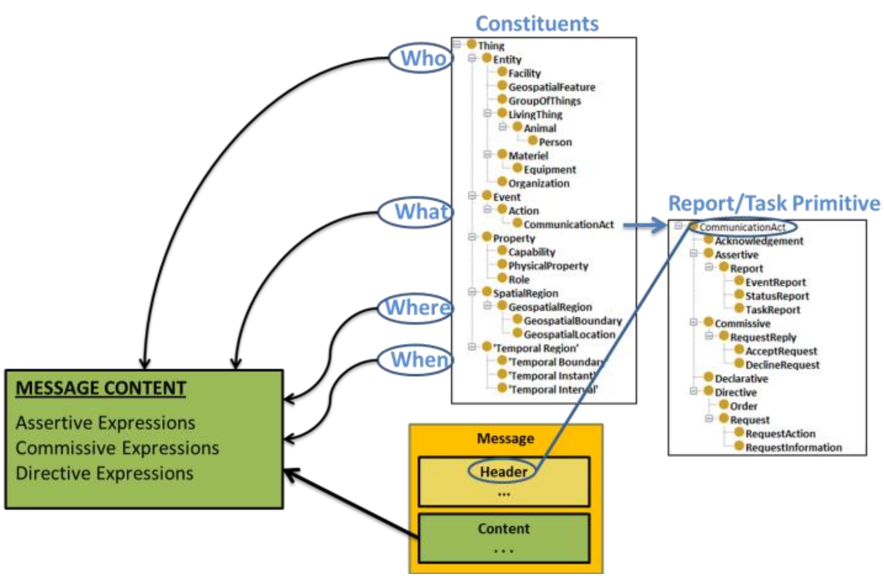

Figure 11 Content Model-Message Framework Relation

Separating concerns in the Content Model and the Message Content allows for multiple complementary, message schemas as opposed to imposing a set of prescriptive formatted text message templates. The decoupling of the content from the message no longer restricts the use to a single XML schema since multiple possible schemas can be aligned to the content model that can be extended, as required.

Supporting multiple schemas allows implementers to meet specific community requirements, without sacrificing interoperability. However, in order to maintain interoperability, it is necessary to ensure that the schemas are aligned. Different schemas may be necessary to meet user requirements for different types of report, order, and request messages. Moreover, having a distinct Content Model better supports aligning C-BML schemas with non-C-BML schemas from legacy models that already may be defined by military specifications, for example. Finally, having a separate Content Model facilitates the semantic alignment of C-BML with other standards, such as MSDL.

Vocabulary and grammar, as captured in the Content Model and Message Framework, are sufficient for many C-BML use cases involving simple message threads (e.g. Report/Acknowledgment). However, for many use-cases or mission threads, larger numbers of related messages typically are exchanged and referenced. To support more complex information flows related to these message threads, more standardization is needed. Interaction Protocols govern the communication of messages related to a specific mission thread, such as Call-For-Fire, as depicted figure 8. Interaction Protocols specify how messages relate to and depend on other messages for a given message thread and dictate the rules that C-BML clients must conform to when participating in a message thread.

The Core C-BML Services fulfill two distinct needs: 1) to provide a standard interface for basic C-BML message operations; and 2) to allow C-BML clients to define their own services based on orchestration and/or extensions of the former. Finally, the Normative Specification layer will include the Core C-BML Services Specification that provides the protocol-specific description in terms of implementable technologies depicted in figure 12. Depending on stakeholder requirements, this could include SOAP, REST, WebSockets, or Server-Sent Events (SSE) for web services; HLA, DIS, or TENA for simulation interoperability architectures; or, SMTP, AMQP, XMPP, or OMG-DDS for enterprise messaging[3].

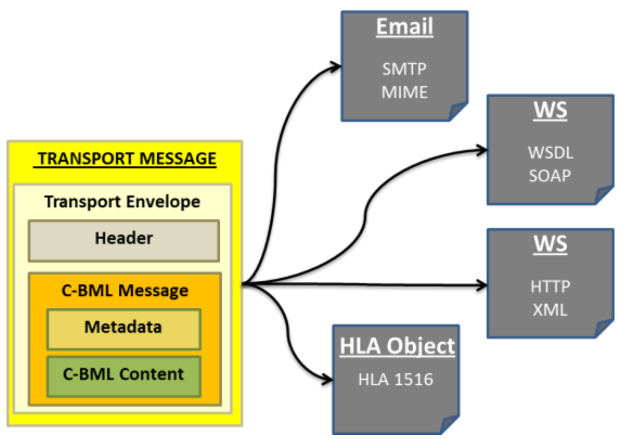

Figure 12 Information Exchange Mechanisms

[3] SOAP – Simple Object Access Protocol
REST - Representational State Transfer
HLA – High-Level Architecture
DIS – Distributed Interactive Simulation
TENA – Test and Training Enabling Architecture
SMTP – Simple Mail Transfer Protocol
AMQP – Advanced Message Queuing Protocol
OMG-DDS – Object Management Group's Data Distribution Service

3.6. Specification Guidance

The Specification Guidance layer of the SDF will encompass sample services, message schemas, and sample data. The guidance may illustrate example community extensions of content models and message frames. The guidance also may illustrate how tactical messages or system-specific messages can be formulated using C-BML's extensibility. Finally, the guidance layer also likely will address technology-specific concerns such as those relating to commonly used programming languages.

3.7. Reference Implementation

The creation of Reference Implementations (RIs) as part of standards development is a common practice, as it provides a chance to prototype, test, validate, and revise the specification under development. C-BML stakeholders have recognized the value that a C-BML RI would bring to the C-BML drafting process, though the conditions for doing so need to be established. The C-BML drafting groups do not currently have a mandate to create a RI and may defer to the PDG for guidance. However, the RI has remains as the last layer of the C-BML SDF since it represents one of the means by which the standard can be validated through concrete tests and use.

Like the Requirements or Reference Architecture, RIs are not part of the normative or guidance specifications for C-BML, but are included as part of the SDF stack as an essential part of the standard development process.

4. Applying the C-BML SDF

Although still in the early stages of development, the C-BML SDF already has been useful in the C-BML Phase 2 standard development activity, as described in the following section.

4.1. Use of Complementary Model Representations

The different SDF layers deals with different, complementary representations of the C-BML model: OWL ontologies, UML models, and XML Schemata. The Core Content Model is represented as a set of OWL Ontologies, the Requirements and Foundation classes are represented in UML while the physical model is represented as XML schemata.

The approach of a complementary use of OWL ontologies and UML models for defining standards is described in reference [22]. Applied Ontology has its roots in Artificial Intelligence while UML originated from the world of software engineering. C-BML is a standard for software applications that deal with machine-computable, parsable messages that are destined, in many instances, for consumption by software agents. Thus, an approach involving the combined use of OWL and UML is also relevant to the C-BML standardization effort. Finally, both OWL and UML provide well-defined mechanisms and tools support for generating derived physical data models expressed as XML schemata.

4.2. C-BML Phase 2 Development

The C-BML SDF has been elaborated as part of the recent C-BML Phase 2 standard drafting activity with the express purpose of facilitating the Phase 2 product development. Toward that goal, an initial instance of the SDF has been created using an UML modeling tool, Sparx Systems Enterprise Architect (EA). Figure 13 is a screenshot of the EA project workspace. The layers of the SDF are present as packages in the UML project. The requirements package utilizes the Requirements Profile extension for UML. Note that the Ontology toolbar (at the left in the figure) allows for the definition, import, and export of OWL ontology elements.

Consistent with the Object Management Group Model Driven Approach, EA supports transforms between the various model representations. It therefore is possible to transform a content model represented as an OWL ontology into a set of XML schemata.

UML Modeling tools such as EA also allow for the automated generation of documentation in the form of web pages or formal documents. This allows easy sharing of model snapshots with a larger community that may not have access to the tool or know how to use UML tools.

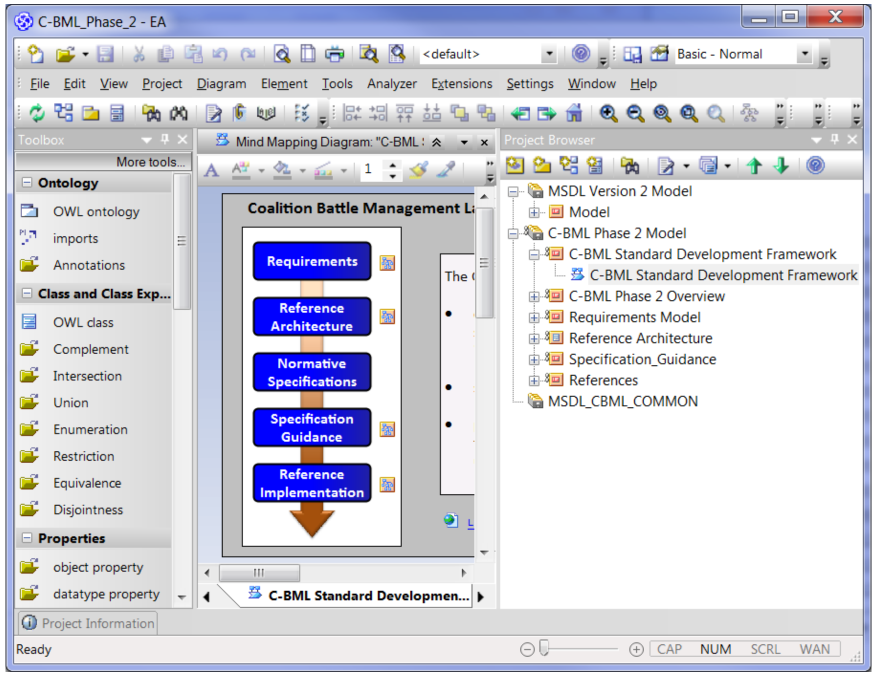

Figure 13 UML Implementation of C-BML SDK

4.3. Using the SDF as a Guide for Extending C-BML

As with any product development, establishing the scope and product development plan is critical to ensuring the products' timely availability and usefulness. Throughout the preceding sections, a common concern has been expressed concerning the diversity of requirements and implementation options due in large part to a myriad of technology options and a vast, heterogeneous community of implementers. Another related concern that was articulated was ensuring that flexibility and extensibility also were present in all SDF layers.

It is expected that as part of the C-BML product lifecycle, changes will be required based on change proposals for extensions from various community implementers. Once the preliminary C-BML products have been released for initial use by stakeholders, the SDF will provide the means for these communities to formulate change proposals for consideration in subsequent versions of the standard.

Community extensions and change proposals will be able to be formulated as applications of the normative specification that reference the normative specification while providing concrete examples of how the standard has been applied and why the change is required.

This approach is similar to that employed by the MIP for processing change proposals as described in reference [23].

The SDF aims to support the needs of multiple communities by providing a means for extending each part of the framework. As an ontology model, the Core Content Model may be extended with domain-specific elements. The message framework may be applied to entire tactical message sets, the results of which may be cataloged for reuse within or across communities. Similarly, interaction protocol definitions also may be cataloged, especially as they relate to mission threads and message threads. Service specifications may result in service implementations, which also may be shared as APIs and SDKs within communities. Furthermore, since the C-BML SDF guides the creation of these extensions, the SDF is the common framework that ensures that all extensions, catalogs, and software related to C-BML are expressed in a consistent manner and thus are aligned.

4.4. Leveraging the Multilateral Interoperability Programme Products

Reference [6] made the following recommendations for a C-BML SDF implementation.

Reuse MIP Products: The Phase 1 products built a vocabulary based on the MIP JC3IEDM although no automatic mechanism was defined for updating the C-BML model following changes to the MIP products. The current SDF instance is being utilized as a means to reuse the MIP Information Model (MIM) specification in C-BML and involves the use and modification of dedicated automation tools provided by the MIP. This work also will facilitate and expedite the creation of future revisions of C-BML following subsequent releases of the MIM. This work is being conducted in collaboration with the MIP Block 4 PIM Working Group.

Automate Model Creation and Maintenance: Establish the procedures and mechanisms to automate the creation and maintenance of the complementary C-BML model representations: C-BML OWL Ontology, C-BML UML Model and C-BML XML Schemata. This work builds on the use of the MIP tools and also may include UML transformations.

Align C-BML and MSDL: Coordination and convergence of the MSDL and C-BML standards activities remains a top priority within SISO. As the respective MSDL and C-BML PDGs deliberate on a way forward, the SDF includes artifacts for addressing semantic (content model) and syntactic (message framework) alignment. Future efforts include defining an automatically generated common core model that can be utilized for defining both MSDL and C-BML products.

Expand the SDF coverage of system integration: For C-BML to support realistic actor-to-actor communication across tactical and simulation networks, the SDF will need to identify design patterns for solving common integration challenges. Coordination with other standards development groups also will be required.

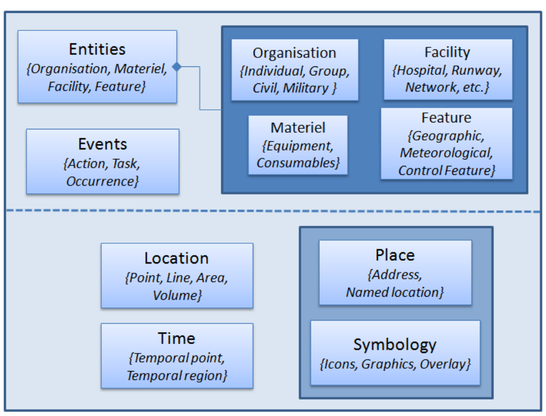

Figure 14 C2-SIM Entities, Events & Properties

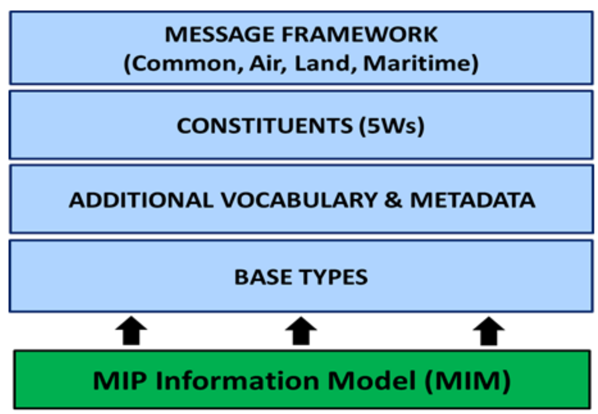

Figure 15 Proposed C2-SIM Logical Model

The first three of these recommendations have been followed in a prototype implementation through collaboration between the MIP Block 4 PIM Working Group and the authors. This collaboration has involved the use of the latest MIP model, the MIM, which is much improved in terms of usability and reduced complexity [17]. Many of the shortcomings of the JC3IEDM have been remediated and furthermore the MIM toolset allows for the rapid creation of subviews that can be derived from the MIM. The toolset also provides the flexibility for the user to re-use any subset of MIM types (i.e. classes, enumerations, core data types etc…) and also to modify these types; to add new types, stereotypes, associations and packages.

The MIM toolset includes a model editor wherein the C-BML core content model can be defined. Also included in the toolset is an XSD generator that creates a set of XML schemata with the C-BML core content model as input. Figure 14 illustrates the domain objects required for C2-to-simulation interoperation in terms of entities, events and properties. This depiction is consistent with the MIP products where "entities" correspond to MIP "objects" and "events" correspond to MIP "actions". The term C2-SIM used here is indicative of the effort to merge military scenario initialization (i.e. MSDL) requirements and military scenario execution (i.e. C-BML) requirements into one unified model.

Figure 15 shows how the MIM-based approach has been used, consistent with the C-BML SDF, to create a layered model. The first layer represents the foundation classes, defined as a pure subset of the MIM. Additional types and metadata are added in the second layer while the third layer defines the composite types, including those referred to as the 5Ws. The first three layers comprise the core content model, while the last layer, the message layer, represents the Message Framework. The results of this work have shown that it is feasible to create a unified C2-Simulation interoperability model for military scenario initialization and execution using the MIM and associated tools.

5. Relation to Architectural Frameworks

As the elements of the C-BML SDF were defined, it became evident that there were many similarities between the SDF and the Canadian DnDAF, the US DoDAF, the UK MoDAF and the NATO NAF architectural frameworks. C-BML by itself is not a program or a system, in the acquisition sense, but instead likely will be specified as a requirement in Request For Proposals for systems and subsequently applied to systems architectures. By following architecture-driven engineering practices, the SDF could provide a profile for the some of the various AF viewpoints, such as the Operational Viewpoints (OV), Capability Viewpoints (CV), Service Viewpoints (SvcV), Data & Information Viewpoints (DIV) and Standards Viewpoints (StdV) to depict more seamlessly how to implement and deploy C-BML solutions.

The following table maps some of the relevant DoDAF v2 views and viewpoints to the SDF. This assessment is preliminary, as the details of the SDF and the C-BML Phase 2 products emerge.

This table illustrates the primary architectural views that likely will be influenced by or require C-BML elements.

Relating the C-BML SDF to these architecture frameworks provides the basis for illustrating C-BML's operational relevance in requirements and design processes and may facilitate including C-BML references in artifacts created using these frameworks.

C-BML SDF Section	DoDAF/MoDAF View
Requirements Model	AVs, CVs, OV-1, SvcV-1
Reference Architecture: Content Model	DIV-1, DIV-2
Reference Architecture: Message Framework	DIV-3, SvcV-6
Reference Architecture: Interaction Protocol	OV-5, OV-6c, SvcV-10c
Reference Architecture: Service Components	OV-2, OV-3, OV-6b, SvcV-2, SvcV-4, SvcV-10b
Normative Specification	StdV-1
Specification Guidance	StdV-1

Table 1 C-BML relation to MoDAF/DoDAF/NAF

6. Summary

The C-BML community is working towards a physical interoperability solution, yet inevitably technology will evolve, stakeholder needs will change and therefore this solution must be agile and easy to modify and track.

The C-BML SDF has been developed for the purposes of aiding the development and communication of the C-BML Phase 2 Products. It also may prove useful to developers as they use the C-BML standard for their implementation purposes. The C-BML Phase 2 DG plans to continue developing and using the C-BML SDF as an integral and unifying element of the drafting activity.

Currently available technology provides opportunities for automating much of the standard development activity by providing the means to allowing for generating specification products instead of manually handcrafting these products, which can be time-consuming and prone to errors for both product development and during subsequent product revisions, as the standard evolves. Automation technology already has successfully been applied in the development of standards products such as those produced by the MIP and preliminary work has shown that the latest MIP product, the MIP Interoperability Model (MIM) and associated toolset provides an excellent basis for defining the C-BML Logical Data Model and derived Platform Specific Models, such as XML schemata. Furthermore, a MIM-based approach is consistent with the C-BML SDK and offers a timely opportunity to proceed with the merging of the MSDL and C-BML standards.

The issues and challenges that motivated the creation of the C-BML SDF also apply to the development of standards, in general. The C-BML SDF approach is well suited, in particular, for managing the development of standards that have strong dependencies on other standards such as those being developed within SISO and by other standardization bodies.

The authors suggest that there is a potential to apply the SDF to other standards. For instance, SISO, NATO, and DoD M&SCO have shown an increasing interest in better cohesion, compatibility, and reuse across standards. The SDF defines a logical partitioning of the data, message, service, interface, and behavior aspects of C-BML. Similar challenges are present in other standardization activities and the SDF provides a starting point for relating, reusing, and aligning otherwise disparate standards.

References

[1] Argo, H., Brennan, E.J., Collins, M.W., Gipson, K., Lindstrom, C., and MacKinnon, S., "Level 1 Model for Battle Management Language (BML-1), TEMO Simulation Laboratory (TSL), Fort Leavenworth, KS, 23 March 1999

[2] C. Blais, K. Galvin and M. Hieb, "Coalition Battle Management (C-BML) Study Group Report", Paper 05F-SIW-041, Fall Simulation Interoperability Workshop Sept 2006.

[3] J. Abbott, S. Levine, M. Pullen: "Answering The Question Why A BML Standard Has Taken So Long To Be Establishes?", Fall 2011 SIW, Orlando USA.

[4] K. Heffner et al., "A Systems Engineering Approach to M&S Standards Development: Application to the Coalition Battle Management Language", 13S-SIW-002, SISO Interoperability Workshop, San Diego CA, April 2013.

[5] B. Lang, M. Gerz, O. Meyer, D. Sim: "An Enterprise Architecture for the Delivery of a Modular Interoperability Solution", NATO RTO-MP-IST-101, Information Systems Technology Panel Symposium, Oslo 2011.

[6] K. Gupton, K. Heffner: "A Standards Development Framework for C-BML Phase 2 and Beyond", Fall 2012 SIW, Orlando USA.

[7] F. Truyen, "The Fast Guide to Model Driven Architecture: The Basics of Model Driven Architecture", Cephas Consulting Corp, January 2006, [On-line] www.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf, last accessed April 15, 2013

[8] K. Heffner: "NATO MSG-119 C2-Simulation Interoperability Workshop Technical Evaluation Report", May 2013.

[9] NATO Allied Procedural Publication, APP-11(C) NATO Message Catalogue, December 2009.

[10] US Army Report and Message Formats, FM 6-99.2, C-1, Sept 2009.

[11] K. Heffner and F. Hassaine, "Towards Intelligent Operator Interfaces in Support of Autonomous UVS Operations", ICCRTS 2011, June 2011.

[12] K. Müller, "NATO and XML", XML Europe Conference, June 2000.

[13] K. Gupton, C. Blais and K. Heffner, "Management of C2 data standards with modular OWL ontologies", International Journal for Intelligent Decision Support Systems 4(3), pages 286-207, October 2011.

[14] M. Ucuncu and M. Demirezen, "Formatted Message Exchange in a Multinational Environment" Information Fusion for Command Support, pp. 2-1 – 2-10). 2006.

[15] NATO MSG-048 (C-BML) Final Report, February 2012.

[16] G. Muller, "A Reference Architecture Primer," Eindhoven Univ. of Techn., Eindhoven, White paper, 2008.

[17] M. Gerz, N. Bau, "A Platform-Independent Reference Data Model for a Future Interoperability Solution", 17th ICCRTS, Fairfax, June 2012

[18] E. Oishi, "Austin's Speech Act Theory and the Speech Situation", Esercizi Filosofici 1, 2006, pp. 1-14.

[19] E. Borgers, M. Spaans, J. Voogd and M. Hieb, "Using a Command and Control Language to Simulate Operations in a Multi-Agent Environment", 13th ICCRTS, June 2008.

[20] Foundation for Intelligent Physical Agents (FIPA) Communicative Act Library Specification, Dec 2002.

[21] J. Bermúdez, A. Goñi, A. Illarramendi and S. Santini, "Is an OWL ontology adequate for foreign software agents communication?", Applied Ontology 2, (2007), pages 351-372.

[22] F. Pinet, C. Roussey, T. Brun and F. Vigier, "The Use of UML as a Tool for the Formalisation of Standards and the Design of Ontologies in Agriculture", Advance in Modeling Agricultural Systems, Springer Optimization and its Applications, Vol 23, 2009, pp. 131-147.

[23] M. Gerz and N. Bau, "A Platform-Independent Reference Data Model for a Future Interoperability Solution", 17th ICCRTS, June 2012.

Author Biographies

KEVIN GUPTON is an Engineering Scientist in the Modeling and Simulation Information Management Group at the University of Texas at Austin, Applied Research Laboratories (ARL:UT). He has over 10 years of experience in enterprise system engineering, data modeling, and knowledge management. Mr. Gupton has developed net-centric applications and data services utilizing C4I and M&S common data standards and ontologies. He holds a Bachelor of Science in Mathematics and a Master of Science in Computer Science from Texas A&M University.

DR. KEVIN HEFFNER holds a BS in Engineering from the State University of NY at Buffalo and a Ph.D. from University of Paris VI. He has worked in the field of modeling and simulation for over 20 years. His work includes applying model-centric concepts to flight simulator architectures and interoperability among C2, simulation, and automated forces, including Unmanned Aircraft Systems.