C2 Product-Centric Approach to Transforming Current C4ISR Information Architectures

Dr. Israel Mayk, and Bernard Goren
US Army Communications-Electronics Research, Development and Engineering Center (CERDEC) C2 Directorate (C2D) Fort Monmouth, NJ

Abstract

The Army Future Force (FF), including Units of Employment (UE) and subordinate Units of Action (UA) equipped with Future Combat Systems (FCS) and manned by Future Force Warriors (FFW), is distinguished from legacy and interim force units and systems by its exceptional responsiveness, deployability, agility, versatility, lethality, survivability and sustainability. These characteristics are meaningful individually but are not completely independent of each other. To support such characteristics, the Future Force is expected to be an order of magnitude more complex in its force structure, doctrinal requirements and technology than the current legacy force. A coherent, comprehensive and manageable set of complementary reference concepts are needed to facilitate the formulation of an overarching information model, reinforced by the rigor of UML, XML and metadata registries, to support system engineering and integration and to assess interdependencies and tradeoffs among the above characteristics based upon more detailed capabilities and their measures-of-performance (MoPs). In this paper we discuss and motivate the use of such an information model derived from the C2RM that is well suited to flesh out C2 UML architectures and C2 XML schemata. The C2RM defines common layers for C2 entities where each layer defines and requests Information Exchange Requirements (IERs) from peer entities using the layer below as well as provide an Information Exchange Products (IEPs) to peer entities which encapsulate the results of its services as requested by the next higher layer. The IERs/IEPs are organized into a common schema following the outline of an operations order (OPORD). This same information model is used to address the problem and offer insight into what it would take to establish a formal language for C2, to derive the rules for analyzing and parsing C2 products from natural language to machine language for use by C2 Applications, and to leverage commercial representation and modeling languages such as the Unified Modeling Language (UML) and Extensible Markup Language (XML and associated metadata registries and tools.

Background

The UML [1] and XML [2] standardization communities are establishing common vocabularies that take advantage of both worlds. In particular, the unifying concept of a Model-Driven Architecture (MDA) under development by the Object Management Group (OMG) leverages the Meta Object Facility (MOF), Common Warehouse Meta-model (CWM) and UML to spin off domain specific reference frameworks in the area of e-commerce, finance, manufacturing, telecom, space, healthcare and transportation. In addition, hundreds of XML standards based upon a core foundation and enriched by the two middleware pillars of message-oriented and document-oriented specifications support vocabularies and applications in the areas that include but are not limited to math, science, legal, education, publishing, finance, marketing, sales, travel, food and human resources. The C2 community is also embracing these standards in key programs. ABCS [3] developers embarked on transforming their Information Exchange Requirements (IER) to an XML environment. FCS [4] developers have embraced both UML and XML for their architectures and MIP [5] too is migrating from IDEF1X [6], which is a database-oriented information model to more powerful UML and XML schemata. The scope selected for the C2 domain, however, is often limited to a given system development or a given set of stakeholders and the pace is bursty and not conducive to collaboration. To more fully reap the benefits of these standards a distributed collaborative approach will evolve to establish an integrated reference R&D environment for C2 that includes a comprehensive information model and more specific information architectures for use in describing the operational, technical and system architectures of various developments. These information model and architectures defined via standard metadata registries for the C2 domain would enable the modeling of various C2 processes across organizational boundaries. In addition, and as a result, the C2 community may also be able to reap significant benefits and evolve with a much greater synergism to build more effective network–centric C2 systems, product improvements and interoperable extensions.

Approach

The proposed information model uses the structure of the C2 Reference Model C2RM [7] as epitomized by the concept of layering services for generic C2 applications and their implementation. Layering is a key design paradigm for managing complexity while promoting updating, maintaining, distributing, and reusing architectural patterns and elements. Data elements are nested hierarchically in accordance with the C2RM layers to provide both syntactic as well as semantic consistency and coherence. Technology/implementation base layers provide the framework for the syntax of C2 products. The syntactic structure is represented by products of topics, components and facts about judgments, recommendations, conclusions, domain objects, modules, signals and energy sources based upon valuable experience, knowledge, information, data, tools, equipment, and supply, respectively. Each technology/implementation base layer provides object-oriented processing, memory, entry, display, and flow services. In an orthogonal dimension (in the same sense that syntax is orthogonal to semantics), application layers provide the framework for the semantics of C2. The semantic structure is represented by missions arising from the full spectrum of conflict and associated plans, tasks, jobs, assignments, transactions and packages which are based upon policy, strategy, tactics, schemata, disciplines, techniques, and instructions, respectively. The C2RM is based not only upon the ISO OSI Reference Model but also upon a general seven-layer meta-model for problem solving. Operational challenges are identified, mapped and framed for each application. Solutions are developed through three subordinate time-dependent layers involving future, present and past timeframe considerations and three subordinate space dependent layers involving multilateral, bilateral and unilateral capabilities. We use the C2RM to describe units and their assets that correspond to the Units of Action of the Future Force. We use the C2 Reference Architecture (C2RA), which leverages upon the C4ISR Architectural Framework to facilitate the transformation of the services identified in the C2RM into Unified Modeling Language (UML)-based layered packages of components that naturally promote network-centric C4ISR-enabled combat systems. Finally we use the C2 Reference Force to enable commanders to tailor UEs and UAs for specific levels of conflict.

With this formulation we proceed to describe how one may go about unifying the FF/UE/UA informational elements that are represented in XML with FF/UE/UA architectural elements that are represented in UML diagrams. Common Natural Language constructs, referred to as W6H, such as: "Who does what to whom when where why and how" are transformed into standard super class diagrams that facilitate object orientation of the design and reflect how objects may collaborate to support a wide variety of use cases to implement decision-support applications. FF/UE/UA architectural elements (found in requirements, design and specification) on the one hand and informational elements (found in C2 products such as OPORDs [8], Decision-aids and tactical messages) on the other are naturally unified through the use of the Extensible Markup Language (XML) and the Unified Modeling Language (UML) standards. Applications are layered using UML packages and subsystems according to the level of conflict, presentation, operation, procedure, network, link, and asset services. UML-defined components, classes and objects that hierarchically process user-oriented, natural language, and semi-structured products provide layered application services. These products include missions, plans, tasks, jobs, assignments, transactions and messages. C2 products are key to federating FF/UE/UA applications and facilitating collaboration and interoperability. A C2 parser may be used to further transform information elements into XML elements and attributes consistent with C2RM-based C2 Markup Language (C2ML) schemas. The C2RM also identifies UML actors who are responsible hierarchically for the execution of a class of methods unique to each layer. Thus, generic commanders, planners, controllers, agents, administrators, coordinators, and operators are actors responsible for UML-based use cases performing analyses and syntheses of policies, strategies, tactics, schemata, disciplines, techniques, and instructions, respectively. Clearly, services that embed human decision-making will require extensive R&D and a high-level maturity of understanding before any agreement may be reached for the purpose of standardization. Nevertheless, the goal for achieving a common framework of terminology, structure and behavior is essential to the success of Army transformation efforts. The research and development efforts of the reference concepts as described in this paper should facilitate this goal.

Introduction

At the foundation of every information model is a set of data elements. All the elements are inter-related using ontology [9]. Ontology is the logical framework or system of terms and their relationships. Typically terms (single words or word concepts) have a taxonomy that places the terms in an information hierarchy in accordance with categories. A model is typically named to represent the logical framework or system associated with a given domain. In general, the structure and behavior of ontologies may be captured in a meta model that establishes the language to be used in the actual model. The meta model itself is based upon or uses a more fundamental structure that represents the language elements or syntax that will contain the domain model. That fundamental structure is also important to understand the domain model and is called the syntactic model. The target domain model is also referred to as the semantic model. Thus the syntactic model is associated with the grammar or structure of the language and the semantic model is associated with the meaning or behavior of the domain terminology.

An information model includes both models. As an example, modeling standards such as IDEF1X or UML provide the syntactic rigor necessary to describe and test data models and object models, respectively, for real-world domain models. The MIP C2 Information Exchange Data Model (C2IEDM) is an example of an IDEF1X application. Object-Oriented implementations are examples of a UML application. Ontology, therefore, provides the information model container or syntactic model. The information model content, or equivalently the semantics model or ontology, fits within the syntactic model hierarchies that are derived from and with the help of a complementary ontology. As another example, in the High Level Architecture (HLA) [10] for modeling and simulations, Object Model Template (OMT) specification is the syntactic model that defines the format and syntax (but not content) of HLA object models. The Simulation Object Model (SOM) and the Federation Object Model (FOM) provide the semantic model or information model content of the simulation domain objects.

Using ISO Standard 11179

Many systems were and continue to be designed without the benefit of a modeling tool that is based upon a formal modeling standard. Nevertheless, one can apply such a tool to reverse engineer and obtain the de-facto syntactic and semantic structures and arrive at the syntactic and semantic models for such systems. By adopting a common but powerful standard in which to define the domain data elements mapping between information models is greatly simplified. ISO Standard 11179 [11] provides the necessary guidance. According ISO 11179, metadata registries are required to identify four interrelated data item types: dataElementConcept (dEC), conceptDomain (cD), dataElement (dE), and valueDomain (vD). It is therefore possible to further clarify and possibly constrain the definitions of these data item types using the six inter-relationships as shown in Table 1.

dEC	←→	cD
dEC	←→	dE
cD	←→	vD
dE	←→	vD
dEC	←→	vD
cD	←→	dE

Table 1. ISO Standard 11179 Key Data Concepts.

It is important to specify relationships a) through d) as a minimum to resolve most if not all practical ambiguities. Given a domain, the language of the metadata registry, for that domain, is specified by identifying all instances pertaining to the four data item types above and their interrelationships. For a given domain, there exists one or more namespace for each data item type above. The namespace includes permissible, recognizable instances of the data types by the user of the metadata registry. We indicate the name space of a data type instance using the double colon symbol "::". A namespace of a subspace of a data item is indicated by the name of the subspace instance that may be followed by the namespace of the space to provide the context. For example:

a. Let dEC:: be any namespace that belongs to a dEC data item.
Let dEC = target, then target is an instance of dEC, and
target:: is the namespace instance of the namespace dEC::.

b. Let cD:: be any namespace that belongs to a cD data item.
Let cD = targetOfOpportunity,then targetOfOpportunity is an instance of cD, and
targetOfOpportunity:: is the namespace instance of the namespace cD::.

c. Let dE:: be any namespace that belongs to a dE data item.
Let dE = targetType, then targetType is an instance of dE, and
targetType:: is the namespace instance of the namespace dE::.

d. Let vD:: be any namespace that belongs to a vD data item.
Let vD = groundVehicle, then groundVehicle is an instance of vD, and
groundVehicle:: is the namespace instance of the namespace vD::.

Note that each namespace instance has its own unique itemValues. We can have many variations for what appears to be and may very well be the same value domain. For example
let vD = groundVehicle, where groundVehicle:: = [M1A1| M2A3| M3A3| M113],
or let vD = grndVcl, where grndVcl:: = [MBT| IFV| CFV|APC]
or let vD = groundVcl, where groundVcl:: = [Abrams| Bradley| Gavin]
Clearly there are mappings between these apparently different vDs. It is only after we define them in the full context of the metaData relationships a)-f) above that we would fully understand the context and applicability. It is possible to have any given vD instance value map to any other vD instance value. It is possible to have each vD instance belong to two different dEs, cDs and therefore different dECs. It is also possible to have two or more vD instances belong to the same dEs, cDs and even the same dECs. The metaData registry IAW ISO 11179 is therefore critical to resolve any ambiguity. Unless such a standard is mandated, the already difficult task of harmonizing and adapting between information models may be compounded since information modelers may not be aware of certain context sensitivities that may accompany certain data elements and attributes.

Using Data Modeling

Data modeling is a fundamental activity of every program to develop an information system. Data modeling is a part of object modeling to identify the objects and their relationships. The data modeling ontology however provides only a very limited syntactic language in which to flesh out complete information models. Data modeling may be used internally to define a syntactic model for a database for use with local or remote client applications. In addition, data modeling may be used to promote information exchange through either a common message exchange mechanism (MEM) or a common data exchange mechanisms (DEM). Information Exchange Data Model is a formal specification of information exchange between collaborating parties within a business area. It must be based on a complete information model since it is used to promote semantic interoperability by facilitating coordination and de-confliction of valid approaches and requirements.

The LC2IEDM which evolved from the ATTCCIS Generic Hub data model describes of battle space objects using terms such as listed in Table 2.

Object	Rule of Engagement
- Organization	Action
- - Unit	- Task
- - Convoy	- Event
- - Post	- Objective
- Facility	- Target
- -Bridge	- Effect
- -Minefield	- Status
- Person	Location
- Materiel	- Point
- Feature	- Line
- - Control	- Surface

- - Geographic	- Volume
- - Weather	Report
Capability	Context
- Maneuver	Reference
- Fire	
- Surveillance	
- Engineer	
- Storage	

Table 2. LC2IEDM Key Semantic Model Data Elements.

Operational architectures, system architectures and technical architectures are insufficient to define (require and specify) an information system. They provide insight into the "body" of a system. The system however can only "come to life" after we provide it with an integrated information architecture. Database developers traditionally adopted data modeling as a method for structuring information repositories and specifically relational databases. Before the emergence of UML, IDEF1X was developed by the US Air Force and adopted by several toolmakers to use as the syntactic model for data structures and business rules. IDEF1X notation uses Entity Relationship diagrams (ERD) in which entities represent real or abstract things just like objects and classes. However, unlike classes, entities do not have methods. As shown in Table 3, they only encapsulate attributes and references called keys. Keys are used to uniquely identify an entity within a model or a database. An entity may have relationships with other entities and be part of a family of entities with a parent and child categories that captures the hierarchy of the information architecture. The C2 domain has evolved through several generations of data models including the C2Core Data Model, C4ISR Architecture Data Model (CADM). The Joint Common Data Model (JCDM) and the C2 Information Exchange Data Model (C2IEDM)

Entity
- Attribute
- - Key
- - Data
- Relationship
- - Dependency
- - Category
- - Verb Phrase
- - - Cardinality

Table 3. IDEF1X/ERD Top Level Syntactic Model Data Elements.

Work is underway to consolidate the C2IEDM developed under the Multilateral Interoperability Programme (MIP) and related data models maintained by the NATO Data Administration Group (NDAG) communities into one Joint/Combined Consultation, Command and Control (C3) Information Exchange Data Model (JC3IEDM).

MIP is chartered to achieve international interoperability of Command and Control Information Systems (C2IS) at all levels from corps to battalion, or lowest appropriate level, in order to support multinational (including NATO), combined and joint operations and the advancement of digitization in the international arena.

Understanding the DIS/HLA Ontology

Simulations are abstractions of the real world, and no one simulation can solve all of the functional needs for the modeling and simulation community. However the power of simulations is brought to the fore when they are networked within the simulation world to address a larger environment as well as with the real world to provide stimulation and feedback. While DIS/HLA constitute an integrated approach that has been developed to provide a common architecture for simulation, their ontologies must map to the C2 system battle command battlespace to enable potential synergy. Although HLA had evolved and replaced the Distributed Interactive Simulation (DIS) applications, their ontologies are very similar. DIS ontology is contained within the syntactic framework of the Protocol Data Units (PDUs) that corresponds to the OMT. However, the mapping of DIS PDUs and HLA OMT structures to C2 systems data and object models remains a challenge. As an example, the Real-time Platform Reference FOM is based upon the DIS Data Dictionary for PDU Data (IEEE 1278.1-1995) and the Enumeration and Bit Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications, IST-CF-02-01, (commonly known as EBV 2002-1).

DIS data dictionary defines an entity tree as follows: An entity consists of a set of Entity Types. This is the value domain (vD) for the data element (dE) called Entity. At the highest level of abstraction an Entity Type is called a Kind. Each Entity Kind is also a dE with its own vD called by a Domain. Each Kind Domain is also dE with its own vD called by a Category. Leaf instances also known as objects of specific classes in each category are further grouped by dEs called Country of Origin, Subcategories, Specific and Extra. The DIS data dictionary refers to naming of elements of each vD as well as to the naming of the leaf instances as an "enumeration." Thus we have a 5-level hierarchy for the syntactic model as shown in Table 4.

Entity
- Kind
- - Domain
- - - Category
- - - - Country
- - - - - Instance

Table 4. DIS/HLA Key Syntactic Model Data Elements.

In the semantic model, there are ten Kinds of Entity Types as shown in Table 5. The Entity Kinds called "Other", "Lifeform", and "Radio" are not defined but may be used as placeholders for real entities that are yet to be modeled. Note that more complex entities may be built by aggregating these fundamental entities. This is done in the C2RM where we define the concept of a unit. For each of the remaining Entity Kinds with the exception of "Munition" and "Supply", there are six standard Domains 0) Other, 1) Land, 2) Air, 3) Surface, 4) Subsurface and 5) Space.

0) Other
1) Platform
2) Munition
3) Life form
4) Environmental
5) Cultural feature
6) Supply
7) Radio
8) Expendable
9) Sensor/Emitter

Table 5. DIS/HLA Top Level Semantic Model Data Elements.

These sub-domains refer to the environment and appear to be problematic in characterizing multi-mode platforms such as amphibious vehicles, seaplanes and other hybrid vehicles/vessels. Under the C2RM, platforms may have multiple transportation ports that provide them with the capabilities to move in each type of environment. For "Cultural Feature" Entity Kind only the "Land" Domain is defined and for "Environmental" Entity Kind only the "Subsurface" Domain is defined. This suggests that these two entity kinds could have been merged because both characterize the physical environment. Similarly the Entity Kinds "Munition" and "Expendable" may be merged as well as the Entity Kinds "Radio"and

“Sensors/Emitter” may be merged because fundamentally all four are just different types of payload packages to platforms. These Entity Kinds map very nicely to the C2RM concept of ports and packages.

Modeling Warfighting Objects

Warfighting Objects are represented by the Common Warfighting Symbology in MILSTD 2525b [12] that provides the visualization objects for applications essential to display the force and the battlespace to warfighters. It evolved from 1994 to current version that is used pervasively in display services for presenting the Integrated Battle Command Picture (IBCP), the Common Operational Picture (COP), the Common Relevant Operating Picture or the User Defined Operational Picture (UDOP). It is a single document that does not include its own data dictionary for the domain model but relies on other references for rigorous definitions of domain terms. However these references are not formal when compared to the requirements of ISO STD 11179. MIL-STD-2525B defines the composition, construction, and display of tactical symbols and tactical graphics. There are five approved symbol set as shown in Table 6.

Units, Equipment, and Installations (UEI)
Tactical Graphics for Military Operations
Weather (METOC)
Signals Intelligence
Mil. Ops. Other than War (MOOTW)

Table 6. MILSTD 2525b Syntactic Model Data Elements Domain.

UEI, Signal Intelligence and OOTW objects include both force domain as well as engagement domain objects that are characterized by warfighting object icon and frame. Battlespace geometry objects such as points, lines and areas characterize tactical Graphics for Military Operations and METOC. The symbol sets provides a listing of symbol identification codes, hierarchy flowcharts, and technical specifications specific to a given set. Each of the symbols can be cross-referenced to an information hierarchy (taxonomy). The information hierarchy provides an organization or structure for domain that encompasses the tactical information commonly exchanged via symbology intended for visually display.

The syntactic data model for MILSTD 2525 is essentially that of a symbol. As shown in Table 7, at the top level, the syntactic data model classifies a domain object symbol as either a Warfighter Object or a Battlespace Geometry Object. Note that the MILSTD 2525 UEI and SI objects correspond to C2RM units object. MILSD 2525 Tactical Graphics correspond to C2RM coordination objects. And NETOC is a subset of the C2RM environment objects. Note that terrain is not included in MILSTD 2525.

Symbol
- Warfighting Object
- - Icon
- - Frame
- Battlespace Geometry Object
- - Point
- - Line
- - Area

Table 7. MILSTD 2525 Syntactic Model Key Data Elements.

Understanding the Ontology underlying the C2 Reference Model

The purpose of a reference model is to provide the necessary ingredients to align and harmonize (de-conflict) different information models (syntactic as well as semantic models) for the same, overlapping or similar domains. The C2RM provides a powerful framework for mapping and evolving C2-oriented information models regardless of the original intent. The C2RM provides a framework for addressing the full spectrum of conflicts in which C2 systems plan and execute their missions. It's a multi-dimensional layered architecture integrated across the key sub-domains to include all battlefield functional areas of C2. C2RM Objects are classified according to the following categories. For the syntactic model we use UML in conjunction with the syntactic meta-model as shown in Table 8 below.

Conflict Region Objects
- Unit Objects
- - Asset
- - - Platform
- - - - Port/Payload
- - - - - Package
- Coordination Objects
- - Frame
- - Geometry
- Environment Objects
- - Space
- -Air
- -Ground
- - - Land
- - - Subterrain
- - - Beach
- - Sea
- - - Surface
- - - Subsurface
Decision-Aid Objects
- C2 Product
- - Document/Message
- - - Topic
- - - - Component (Table/Overlay)
- - - - - Fact
- C2 Application
- - Service
- - - Utility
- - - - Facility

Table 8. C2RM Key Information Model Data Elements.

Decision-aid Objects are developed as part of C2 applications using UML and object-oriented programming (OOP) to describe the Conflict Region and its objects in the context of the decisions that the user is required to make given his/her mission and relevant facts about the situation. C2 applications structured as a collection of services, utilities and facilities operate on the domain Conflict Region Objects to create, process, display, store and disseminate C2 Products. An XML schema that provides a hierarchical syntactic container for domain-oriented data elements captures the information model for C2 products.

Developing an XML schema for C2 Products

An XML representation for C2 should be derived from the data elements and value domains defined for the corresponding C2 information model including elements from both syntactic and semantic structures as described above. General rules should be established for naming XML elements and attributes and specifying a dictionary template. For C2 products, we use the following naming convention:

1) Tag names that are acronyms are composed totally of upper case letters.
2) Tag names that are not acronyms begin with a lower case letter.
3) Tag names are generally devoid of vowels.
4) Tag names can be formed by concatenating words together as follows:

<tag> = <word1Word2Word3…Wordn>

5) Tag names that are formed from concatenated words are capitalize in the beginning of each word with the exception of the very beginning word and a word that comes after an acronym or a number.

6) Tag names that use consecutive acronyms use a lower case letter for the next acronym.

The general rules for specifying the C2 information elements is as follows:

First we show the XML statement in which the XML elements and attributes are used.

We then define the data elements (dEs) and the value domains (vD) used to specify the data elements. A dE "tag"==> tag: is designated with a single colon postfix following the name of the data element, i.e."tag". For example if tag = unit, then dE = unit:. A vD "tag"==> tag:: is designated with a double colon postfix following the name of the value domain, i.e."tag". For example if tag = size, then vD = size::. Prior to applying XML to data elements, we note that any tag: can represent either an XML element or an XML attribute. tag::, the corresponding vD for tag:, fully specifies the representation, format and range of values allowed for tag:. Nested tags are concatenated in a top-down way, i.e., for tag1:tag2: = tag1Tag2:, tag2: is nested within tag1:. For example, unit:size: = unitSize: can mean either that size: is a sub element of unit: or that size: is an attribute of unit:. A vE is not a tag but is represented as "tag:::" , i.e., it is the value element (vE) of vD = tag:: and therefore is an instance of dE. For example, size::: = Bn.

Thus given tag1 and tag 2, we form dE = tag1Tag2: with the following definition of a complex dE, tag1:tag2:, pointing to an instance of tag2: associated with tag1: (if applicable). XML allows the structuring of two related data elements using unbalanced and balanced tagging as follows:

For unbalanced tagging:

```
<tag1 tag2="tag1Tage2:::"/>
<unit size="Bn"/>
```

For balanced tagging:

```
<tag1>tag2:::</tag1>
<unit>Bn</unit>
```

The use of either an unbalanced or balanced tagging is not only an implementation issue but also an ontological one. For example key object attributes may be normally assigned as XML attributes. Hierarchical, associated and subordinate objects may be naturally assigned as XML elements.

Consider the general case where a dE represents an object. Objects typically are associated via “is a” and “has a” relationships. As a general rule, the dEs used to identify the object or represent the “is a” relationship are used as XML attributes. The dEs used to quantify the object or show its structural composition via “has a” relationship are nested as XML elements. The object may be one of several types and may have a variety of representations. The key meta data used as XML attributes to specify most XML element objects include the following attributes: type, designation, name, id, representation and format.

Consider the following excerpt from an example OPORD:

> This operation focuses on rapidly occupying the brigade zone to establish US presence, to begin suppressing GSPF insurgents, and to prepare for the defense of KAZAR. The battalions will move immediately to their AOs rather than waiting for the entire IBCT to arrive. Secure PRISTINA and maintaining freedom of maneuver throughout the zone are imperative throughout this operation. Initial aims are to establish full situational understanding and to put the GSPF on the defensive. Dominate the 2nd Battalion zone, neutralizing the GSPF, and setting conditions for a successful defense. The desired end state is full control of KAZAR by its legitimate government.

Note that there is no way to know which part of the OPORD it came from unless it was encapsulated by the appropriate syntax, i.e., <cmp id=“3.a” name=“Commander’s Intent”><cmp> . If two users need to collaborate on the commander’s intent, there is no way to know whether this instance of the commander’s intent is the original one or some subsequent version. The dV for id = “3.a” only provides a general index to locate all versions of the Commander’s Intent, but does not identify a specific one. We therefore add an attribute called dsg to make this component unique over time as this OPORD evolves through the collaboration process. We indicate the type of version by appropriately encoding the vD for cmpDsg:. This saves the need to generate another attribute.

In general components are decomposed into facts. As a general rule every Natural Language (NL) statement is a candidate fact. Complex NL statements are decomposed into as many simple control language (CL) statements that are then represented by xml elements called facts (<fct>). Prior to fact decomposition, we identify all readily recognizable W6H elements that evolve from and extend the foundational Battle Management Language (BML) [13]. For illustration purpose, we apply a color scheme as shown in Table 9 below:

Who /whom/whose:	unit (organization), resource, asset, individual
Which (object/product):	platform, equipment, supply, system, package (image, msg, ordnance, cargo)
What (do):	action, plan, operation, task, mission, results, status, outcome, activity
When (on):	datetime, event, before, after, during, parallel, sequential
Where (at):	place, vicinity, coordinates, region, location, position
Why (to):	purpose, goal, objective
How (by) :	organization, formation), command relationship, timing

Table 9. W6H Key Information Model Data Elements.

Rule 1: Look for a name or reference to a country, organization or a unit. If the name also refers to an area append name with "area." Categorize as to allegiance: friendly unit, enemy unit, host unit, neutral unit, unknown unit. Replace in original paragraph: Note that all the units identified by this rule, {i.e., Brigade, US, GSPF, KAZAR, battalions, IBCT, GSPF, 2nd Battalion, GSPF , KAZAR, legitimate government} are defined in the Task Organization or Situation topics of the OPORD.
Rule 2: Look for a name or reference to an area. Append name or reference with "area:" {zone, area, AOs, PRISTINA, zone, zone, area.} Similarly as for Rule 1, all location objects are found in the Operational Overlay or Situation topics of the OPORD.
Rule 3: Identify words belonging to a single W6H concept domain phrase/value domain phrase and classify them accordingly.
Rule 4: Expand any reference to a unit with the names of the units available from Annex A(for friendly) and Annex B(for enemy).
Applying Rule 1:

<cmp id="3.a" name="Commander's Intent">
Commander's Intent
This operation focuses on rapidly occupying the brigade zone to establish US presence, to begin suppressing GSPF insurgents, and to prepare for the defense of KAZAR. The battalions will move immediately to their AOs rather than waiting for the entire IBCT to arrive. Secure PRISTINA and maintaining freedom of maneuver throughout the zone are imperative throughout this operation. Initial aims are to establish full situational understanding and to put the GSPF on the defensive. Dominate the 2nd Battalion zone, neutralizing the GSPF, and setting conditions for a successful defense. The desired end state is full control of KAZAR by its legitimate government.</cmp>

Applying Rule 2:

<cmp id="3.a" name="Commander's Intent">
Commander's Intent
This operation focuses on rapidly occupying the friendly unit area to establish a friendly unit presence, to begin suppressing enemy unit insurgents, and to prepare for the defense of host unit area. The friendly units will move immediately to their area rather than

waiting for the entire friendly unit to arrive. Secure area and maintaining freedom of maneuver throughout the area are imperative throughout this operation. Initial aims are to establish full situational understanding and to put the enemy unit on the defensive. Dominate the friendly unit area, neutralizing the enemy unit, and setting conditions for a successful defense. The desired end state is full control of host unit area by its host unit.
</cmp>

Applying Rule 3 and 4:

<cmp id="3.a" name="Commander's Intent">
Commander's Intent
This_operation [IAW this OPORD] focuses on rapidly occupying the_brigade_zone to establish US presence, to begin suppressing GSPF insurgents, and to prepare for the defense of KAZAR. The_battalions [1st IN Bn, 2nd IN Bn, 3rd IN Bn, RSTA Sqdrn] will move immediately to their_Aos [AO of 1st IN Bn, 2nd IN Bn, 3rd IN Bn, RSTA Sqdrn] rather than waiting for the entire IBCT to arrive. Secure PRISTINA and maintaining freedom_of_maneuver throughout the_zone are imperative throughout this_operation [IAW this OPORD]. Initial aims are to establish full situational_understanding and to put the GSPF on the defensive. Dominate the 2nd_Battalion_zone, neutralizing the GSPF, and setting conditions for a successful defense. The_desired end_state is full control of KAZAR by its_legitimate_government. </cmp>

Applying all the rules we obtain the following XML instance:


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 4 U (http://www.xmlspy.com) by Bernard Goren (US Army) -->
<cmp id="3.a" name="Commander's Intent" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SINCEprdC2030804.xsd">
<on type="rcv">
  <event type="pkg">OPORD</event>
  <unit name="1st IBCT" aff="US"/>
  <do type="occupy">
    <by type="pace">rapidly</by>
    <at type="zone" name="brigade"/>
  </do>
  <to type="msn">
    <outcome type="presence"/>
  </to>
  <unit aff="US"/>
  <to type="do">
    <do type="suppress"/>
    <unit name="GSPF">
      <asset type="insurgent"/>
    </unit>
  </to>
  <to type="prepare">
    <do type="defend"/>
  </to>
  <at type="region" name="Kazar"/>
  <unit name="*/IBCT" aff="US">
    <unit name="1st IN Bn/IBCT" aff="GE"/>
    <unit name="2nd IN Bn/IBCT" aff="US"/>
    <unit name="3rd IN Bn/IBCT" aff="US"/>
    <unit name="RSTA Sqdrn/IBCT" aff="US"/>
  </unit>
</on>
<on type="arrival">
  <at type="area" name="Skopje">
```

```
    <crd type="APOD"/>
  </at>
  <unit name="*/IBCT"/>
  <do type="move">
    <by type="rule">separately and no waiting</by>
  </do>
  <to type="do">
    <do type="secure"/>
  </to>
  <at>
    <city name="Pristina"/>
  </at>
  <to type="do">
    <do type="maneuver">
      <by type="freely"/>
    </do>
  </to>
  <at type="zone" name="IBCT"/>
</on>
<on type="phase">
  <PoO type="initial"/>
  <to type="do">
    <do type="SU"/>
  </to>
  <to type="outcome">
    <unit name="GSPF"/>
    <do type="defend"/>
  </to>
</on>
</cmp>
```

(Note that we use the following convention:
*/unitName => any subordinate unit of unitName as identified in the Task Organization (TO).
*/phaseName=> any time period of phaseName as identified in the Concept of Operation (CoO.)

The above example is intended to provide an insight into the process of arriving at an XML schema for a C2 product. In addition, it is offered as a benchmark for use by other methodologies to transform NL to XML. The OPORD is perhaps the most difficult C2 Product to parse since it includes a significant amount of NL. Other C2 products such as Position and SPOT reports are much more straightforward. The challenge however is to make all types of C2Products compliant with the single schema. Such a schema was used and demonstrated in the Simulation and C2 Information Systems Connectivity Experiments (SINCE) Program as part of Experiment 1a [14].

Conclusions

C2 information models need to be harmonized across the full spectrum of operations in a unified seamless ontology and schema in support of the general evolution of the current legacy force and the future force. C2 information models need to be applied consistently to display of warfighting objects, environment objects and tactical graphics, to structured message standards, databases and repositories and to collaborative and interoperable decision-support applications. The C2 domain is inherently Object-Oriented and UML is a viable and robust meta-model for C2 architectures and applications from a syntactic view. C2RM is needed as viable and robust meta-model for all C2 domain semantic views of UML models and applications and all C2 XML representations spanning ground, sea, amphibious, air and space operations as well as Joint, coalition, intra and inter service and for Home Land Security. C2 metadata registries will be more effectively utilized if they are designed to correspond to a robust coherent and well-organized C2 meta-model such as the C2RM.

Acknowledgment

The authors would like to acknowledge our sponsors in the CERDEC ILIR and SINCE programs. In particular, the SINCE Program Manager, Dr. Dirk Klose was highly supportive in providing valuable feedback and encouragement and enabled the use of our schema in the SINCE program. In addition we would like to thank the SINCE program developers from C2D (Andrew Chan, and Mike Mai), Gestalt (Kevin Sheehan), Viecore (Lynn Kershner), MAK (Nils Reker), AIS (Erik Eilerts) and LNK (Parag Talekar and Dr. Ravi Kulkarni for implementing the schema in their adaptor modules, to Frank Farance for meta-data modeling support and to our Exchange Engineers, MAJ Avi Yariv (IDF) and Alfred Schloesser (GE) for valuable UML and XML analysis and design contributions.

References

1. UML - http://www.omg.org/technology/uml/index.htm
2. XML - http://www.w3.org/XML/
3. ABCS – http://www.fas.org/man/dod-101/sys/land/abcs.htm
4. FCS – http://www.fas.org/man/dod-101/sys/land/fcs.htm
5. MIP - http://www.mip-site.org/home.htm
6. IDEF1X - Thomas A. Bruce, Designing Quality databases with IDEF1X Information Models, 1992 Dorset House Publishing, New York, NY 10014
7. C2RM – The Command and Control Reference Model for Modeling, Simulations, and Technology Applications, Draft 7, 1994, Israel Mayk, D.Eng.Sc. CECOM-TR-94-1, Fort Monmouth, NJ
8. MDMP - FM101-5 Staff Organization and Operations, HQ, Department of the Army, (in PDF format)
9. Ontology – http://www.jfsowa.com/ontology/index.htm
10. HLA - https://www.dmso.mil/public/transition/hla/
    http://www.sisostds.org/
11. ISO Standard 11179 - http://metadata-stds.org/OpenForum2003/
12. MIL-STD-2525B, Common Warfighting Symbology, 30 January 1999, DoD
13. BML - http://www.uxxi.org/docs/c4i-repository/BML.pdf
    http://www.uxxi.org/c4i.html
14. Mayk, Israel and Klose, Dirk "Experimenting with C2 Applications and Federated Infrastructures for Integrated Full-Spectrum Operational Environments in Support of Collaborative Planning and Interoperable Execution", Proceeding 2004 Command and Control Research and Technology Symposium (CCRTS), June 15 – 17, 2004, SAN Diego, CA.