10th International Command and Control Research and Technology Symposium
The Future of C2

Building An Ontology For Command & Control

C2 Policy Track

Raymond J. Curts, Ph.D., (CDR, USN Ret.)
CommIT Enterprises, Inc.
Arlington, Virginia
raymond.curts@commitent.com
(703) 731-0301 (cell)

Douglas E. Campbell, Ph.D., (LCDR, USNR-R, Ret.)
Syneca Research Group, Inc.
Washington, D.C.
dcamp@syneca.com
(703) 627-4257 (cell)

Building An Ontology For Command & Control

Raymond J. Curts, Ph.D., (CDR, USN Ret.)
CommIT Enterprises, Inc.
Arlington, Virginia
raymond.curts@commitent.com
(703) 731-0301 (cell)

Douglas E. Campbell, Ph.D., (LCDR, USNR-R, Ret.)
Syneca Research Group, Inc.
Washington, D.C.
dcamp@syneca.com
(703) 627-4257 (cell)

ABSTRACT

The definition of "Command & Control" is still being debated within the Department of Defense and a consensus has yet to emerge. As historically shown, striving for a common language, or a common lexicon in any domain tends to be difficult at best. The authors find that the problem may be that we are wrestling with various "Command & Control" definitions rather than discussing the environments in which "Command & Control" exists. So, rather than trying to define the term "Command & Control," the authors believe we must focus on the environments in which "Command & Control" functions exist. Once the environments are defined and understood, then the boundaries and limitations of "Command & Control" also come into focus. These borders are defined by the "domain of discourse," (in this case, the concepts, classes, or Ontology *of "Command & Control"). We need to build such a construct. The authors contend that the domain of "Command & Control" does not require a hard and fast definition, per se, but is in need of an ontology to identify the content and boundaries. The role of ontology is to provide a better structuring of what "Command & Control" is or is not, and to help index and retrieve domain-related information. This paper proposes to: describe and promote the understanding of how ontology relates to "Command & Control" and possibly begin to establish a basis from which an interoperable Command & Control ontology can be developed. Finally, and probably more importantly at this stage, this paper should open the dialogue for further discussion on ontological constructs and their applicability to the Command & Control domain.*

1.0 INTRODUCTION

What is Command & Control? The definition continues to be debated within the Department of Defense and a consensus has yet to emerge. As simply as possible, Command & Control can be defined as the actual process of directing and controlling forces. It is the authority that a commander exercises over his subordinates by virtue of his rank or assignment. A generic Command & Control process is depicted in Figure 1 below [IWIP, 1996].

As defined in JCS Pub 1-02, Command & Control is the exercise of authority and direction by a properly designated commander over assigned forces in the accomplishment of the mission. Command & Control is performed through an arrangement of personnel, equipment, communications, facilities and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission [JP 1-02, 1994].

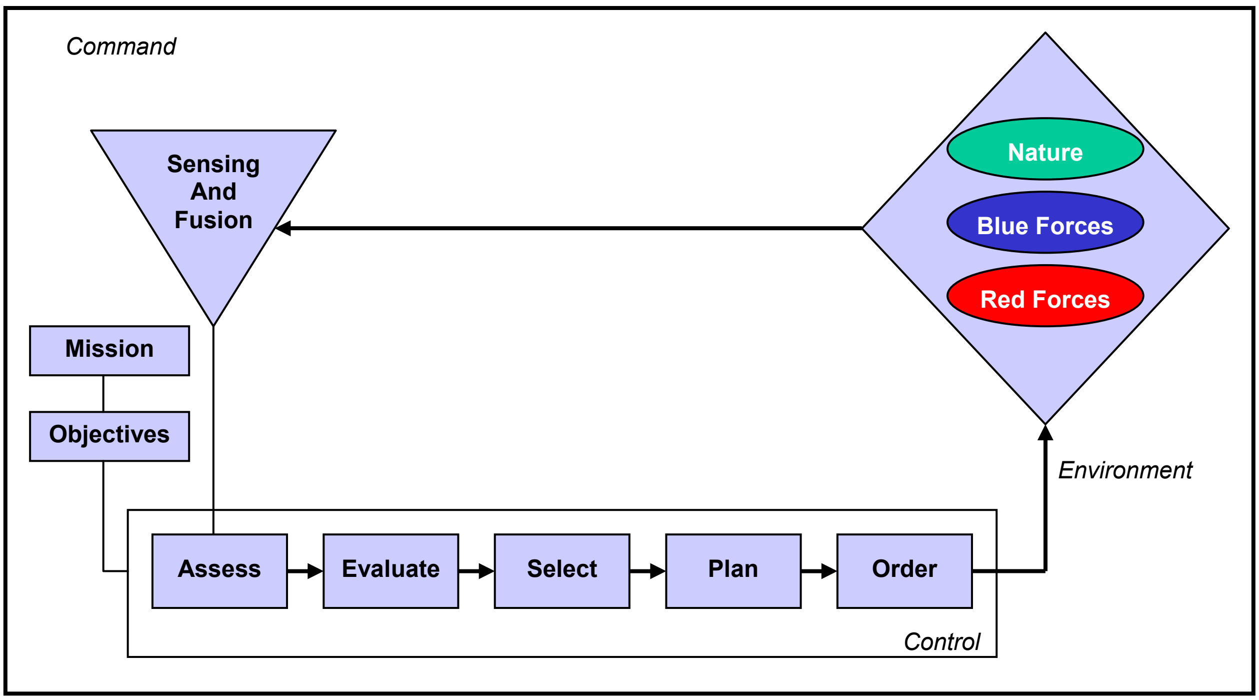

Figure 1: Command & Control Process.

To achieve information superiority, the commander must be capable of making use of the underlying command, control and communications infrastructure in all operational stages from concept through planning, modeling & simulation, to execution in an actual operational environment. Information systems designed to aid in decision-making are commonplace in Command & Control operations and the ability to build, operate and maintain such systems is crucial to the effectiveness of Command & Control. What is needed is the ability to establish a solid ontology for Command & Control decision-making based upon a rigorous, standardized domain definition allowing a wide variety of analyses and acquisition planning.

What is an ontology? According to many, an ontology is defined as a formal and explicit representation of a shared conceptualization [Gruber, 1993]. After some research, the authors' found that there are many different definitions of the word ontology, and some even contradict one another. So, for the purposes of this paper, an ontology is defined as a description of concepts, objects and relationships within a domain and the relevant attributes of each concept, including their restrictions. An ontology, together with a set of individual explicit instances of objects or concepts, constitutes a knowledge base. That is, objects, concepts and their relationships within a domain are the main focus of an ontology.

The reason there are so many different definitions of ontology is because there is no one "correct" methodology for developing one. It is an iterative process: starting with a rough first pass at defining the relevant concepts that make up an ontology. The ontology is then revised and refined as the details are added. In summary:

1) There is no one correct way to model a domain - there are always viable alternatives. The best solution almost always depends upon the application and anticipated extensions.

2) Ontology development is necessarily an iterative process.

3) Concepts in the ontology are primarily objects (physical or logical) and relationships within the domain of interest. In general, *nouns* become *objects*, *verbs* generally indicate *relationships* and *adjectives and adverbs* become descriptive *attributes* of those objects and relationships.

Deciding to construct an ontology for Command & Control, and determining the required level of detail will guide many of the modeling and acquisition decisions down the road. Among several viable alternatives, we will need to determine which one would work better for the projected task, be more intuitive, more extensible, and / or more maintainable. We also need to remember that an ontology is a model of reality within a specific real world domain and the concepts in the ontology must reflect this reality. After defining an initial version of the ontology, it can be evaluated and debugged by using it in applications or problem-solving methods or by discussing it with experts in the field, or both. As a result, the initial ontology will, no doubt, continually evolve. This process of iterative design will likely continue through the entire lifecycle of the ontology [Noy, 2001].

2.0 HOW ONTOLOGY RELATES TO COMMAND & CONTROL

Research from the military community, including recent papers from previous Command & Control Research & Technology Symposia [Alberts, 2003; Auger, 2004; Bourry-Brisset, 2000; Chance *et al*, 2003; Chaum, 2003; Curts *et al*, 1999; Gauvin *et al*, 2004; Gouin *et al*, 2003; Haglich, 2000] clearly demonstrates the many needs and uses for ontologies (and taxonomies) within the Command & Control domain.

Ontologies for Command & Control systems can be instrumental in establishing a Common Operational Picture (COP) among units by making domain analysis, situation analysis and assumptions more explicit. Agents assisting commanders with the Command & Control task will have the ability to "interpret" data and know its meaning and value based upon the domain ontology [Auger, 2004].

One of the most important uses of a well defined ontology is the ability to define boundaries, both internal and external, and analyze the relationships within and across them. In Command & Control community today, one could argue that the internal boundaries are more complex, and perhaps more important, than the external boundaries. Responsibilities for Command & Control within the U.S. Department of Defense, for example, are fragmented across multiple organizations creating internal boundaries that are, at least to some extent, artificial. For example, strategic / global / national Command & Control is largely the purview of the U.S. Strategic Command (STRATCOM) while Joint Forces Command (JFCOM) is generally responsible for theater / regional / tactical Command & Control. In addition, other Command & Control enclaves exist in the areas of nuclear Command & Control, missile Command & Control, logistics and a host of others. There is also a need for ontologies describing coalition operations, functions and the Command & Control systems upon which they reside. These ontologies are essential due to the mix of equipments, operational procedures and computers systems that are involved.

Future coalition Command & Control information systems will have to take into account interoperability issues so that information can be effectively shared and exploited within coalition operations. In this context, interactions between participants require mechanisms to facilitate the exchange of information and provide a shared understanding of the situation based upon common terminology, as a minimum. One solution to facilitate the communication between agents is to build a common ontology that represents a shared model of a domain [Boury-Brisset, 2000].

3.0 ESTABLISHING A BASIS FROM WHICH AN INTEROPERABLE COMMAND & CONTROL ONTOLOGY CAN BE DEVELOPED.

The key word here is "interoperable." Unfortunately, system-specific models often create unsolvable interoperability problems. The result is the Tower of Babel. For example, various systems may report different answers to the question "Where is the enemy aircraft?" One will provide a height above sea level, another height above the ground; one may use latitude / longitude while another provides GPS coordinates; one produces a rho-theta (arc off of a radar sweep) as opposed to a set of numbers and letters based on a grid pattern, and so on. In addition, all of these systems could provide data in either feet or meters. There are no completely unambiguous translations between the systems. And, even with very accurate conversion algorithms, the resulting, fused data is not sufficiently accurate to program today's precision weapons. Shared understanding now becomes limited when all this information is lost or cannot be unambiguously exchanged between C2 elements.

Interoperability comes when a ubiquitous model consisting of a shared vocabulary (i.e., shared meanings or semantics and shared syntax or format) and associated meta-data (i.e., the grammar that defines logically how the vocabulary elements can be used) is in place. In solving the problem of the previous example, engineers have often gone down the wrong path by attempting to improve Command & Control by directly interfacing all these functional systems. This "N-squared" approach is a brute force assault on the Tower of Babel problem, relying on point-to-point information exchanges and translations between the "N" systems. While is bottom-up approach to resolving the problem may have merit, a top-down generic ontology is also needed.

An interoperable ontological construction is a complex collaborative process that crosses individual, organizational, and geographic boundaries. It involves several types of groups with differing expertise, goals, and interactions. An ontology server must be carefully structured to support this complexity. Consider, for example, the task of building schema to support the Command & Control process.

A common Command & Control schema, such as the one being developed as part of the Defense Advanced Research Projects Agency (DARPA) Joint Task Force-Advanced Technology Demonstration (JTF-ATD), would provide a substrate for numerous applications in planning, logistics, intelligence, and so on. With the proper underlying technology, it could support advanced knowledge-based applications as well as conventional databases and software systems. To construct this schema, small groups of experts in each of the key sub-areas collaborate to specify ontologies describing the essential concepts, their properties, and interrelationships. The products of these groups of authors must be merged and checked for consistency by a supervisory board of editors. The editors must then invite comments from a large group of reviewers and critics that include expert peers, end users, and application developers.

As portions of the ontologies stabilize and the editors release them from the reviewing process, larger groups of application developers must become familiar with them and incorporate them into existing, as well as emergent applications. Furthermore, the developers need support to convert the ontologies into a form that they can readily work with in a specific knowledge representation language, database schema language, interface language, or programming language, and they need support for extracting domain models from the ontologies that can be used by problem solving modules [Fikes, 1997].

Thus, the holistic operational transformation sought by the Department of Defense (DoD) requires a complementary transformation in the way that we design, and implement systems. A traditional system engineering technique has been the N-squared approach. An N-squared diagram would represent Command & Control as a set of subsystems, components or processes and their interconnections. The definition of the subsystems and the associated interconnections would be critical to the development of a Command & Control "system" [Percival, 1995]. However, we must move from the N-squared approach to a 1-to-N approach where all share a common "language" defining the domain of interest. That is, a 1-to-N approach is an iterative process where a Command & Control problem would be solved by iterating recursively through a set of commands.

This might be somewhat sobering to those that expected that all interoperability issues can simply be handled with a clever unique interface. To the contrary, our future lies in the ability to evolve to systems and processes that share an operational context model based on a common ontology. So, how might we represent and share operational context?

Today, operational context is initially determined through a top-down planning process. Each echelon effectively adds detail to the Operations Order, Daily Intentions message, etc. At each level watch-standers receive this common guidance as general knowledge and then revisit the specific details most relevant to his or her functional responsibilities. The passing of operational context to our warfare systems must be similarly streamlined. Context must be entered or generated once and shared, each system distilling what is relevant to its function or tasking. Thus, the operations planning process / system outputs should be direct scenario inputs to M&S systems, and the selected course-of-action should be directly readable and executable through the Command & Control and tactical systems. We must ensure that the warfighter is not a data entry clerk for our automated systems. During tactical execution operational context will change and we will need efficient, low / no workload, methods to share these changes in an automated manner across the Naval, Joint and Coalition forces.

To achieve the NCW transformation, sharing operational context and achieving ubiquitous interoperability demand we work top-down, from a shared ontology. If you stop to think about it, this derived requirement is in fact consistent with the best practices within the XML community. That is, XML works best when a functional community is formed and interested participants agree to share a namespace and its associated ontology. Additionally, and perhaps more importantly, as a basis for true interoperability we need an ontology that is country, service, process, system, application, technology, and contractor independent. That is, generic, appropriate to all and specific to none. In the Joint and Combined arena we can not rely on identical hardware and software to enable / ensure interoperability, rather we must rely on system-independent information exchange specifications.

Within DoD we have pursued functional data managers, effectively working to implement system-independent functional definitions supporting information exchange. For tomorrow we require a new baseline, defined at the enterprise level that is functionally generic and system independent. Functional stovepipes preclude composing an integrated representation of military operations and in general operational context. Thus, we must seek a new information exchange standard that addresses the broad scope of battlespace information / operational context we have been discussing [Chaum, 2003].

4.0 A METHODOLOGY FOR CONSTRUCTING A SIMPLE COMMAND & CONTROL ONTOLOGY

Typically, the goal of building ontologies is to create a logical framework, a philosophy, a classification, or to develop a common understanding in a discipline. Creating an ontology intended only to provide a basic understanding of a [Command Control] domain may require less effort than an ontology intended for supporting formal logic arguments and proofs in a [Command & Control] domain [Prieto-Diaz, 2002]. The authors' would like to present a methodology for constructing a simple Command & Control Ontology consisting of a 7-step process as follows:

Step 1. Determine the domain and scope of the ontology
Step 2. Consider reusing existing ontologies or ontology segments
Step 3. Enumerate important terms in the ontology
Step 4. Define the classes and the class hierarchy
Step 5. Define the properties of classes – attributes
Step 6. Define the facets of the attributes
Step 7. Create instances of the objects comprising the ontology

Step 1. Determine the domain and scope of the ontology

Since an ontology is, by definition, a description of a particular domain of interest, we must define that domain and its scope or depth. To do this we must answer several basic questions:

- What is the domain that the ontology will cover?
- What is the intended purpose / use of the ontology – high level, broad scope description or detailed functional breakdown? In other words, what types of questions do we expect the ontology to answers?
- Who will use and maintain the ontology?
- How will it be captured – knowledge base, specialized ontology tools, pencil and paper, etc.?

The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model.

Consider the ontology of Command & Control. Representation of Command & Control objects, functions and processes is the domain of the ontology. We plan to use this ontology to develop a better understanding of how Command & Control is currently conducted within DoD, discover areas that may be of concern and analyze a range of potential policies, processes, procedures and acquisition strategies.

If the ontology we are designing will be used to assist in natural-language processing of reports, it may be important to include synonyms and part-of-speech information for concepts in the ontology. If the ontology will be used to help decision-makers, we need to include all types of information that could influence any particular decision. Determining the relationships between data, information, knowledge, awareness, the environment, decision-making processes and human understanding are all part of developing the ontology. It is important that the people who develop and maintain the ontology describe the domain in a common language (using common terms with accepted meaning within the domain) consistent with the language of the ontology users. Otherwise, a mapping between the languages will be required and confusion, no doubt, will ensue.

One method to determine the scope of the ontology is to develop a list of questions that we would expect Subject Matter Experts (SMEs) to answer, i.e., competency questions [Gruninger, 1995]. These questions will serve as the litmus test later: Does the ontology contain enough information to answer these types of questions? Do the answers require a particular level of detail or representation of a particular area? These competency questions are just a sketch and do not need to be exhaustive.

Step 2. Consider reusing existing ontologies or ontology segments

It is almost always worth considering what someone else has done and checking to see if we can refine, adapt and / or extend existing sources for our particular domain and task. Reusing existing ontologies may be a requirement if our system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies. Many ontologies are already available in electronic form and can be imported into a standardized ontology-development environment. The formalism in which an ontology is expressed often does not matter, since many knowledge-representation systems can import and export ontologies. Even if a knowledge-representation system cannot work directly with a particular formalism, the task of translating an ontology from one formalism to another is usually not a difficult one. However, careful consideration should be given to this issue since ontology languages are evolving quickly and some provide far more utility than others.

There are libraries of reusable ontologies on the Web and in the literature. For example, we could investigate the Ontolingua ontology library (http://www.ksl.stanford.edu/software/ontolingua/) or the DARPA Agent Markup Language (DAML) ontology library (http://www.daml.org/ontologies/). There are also a number of publicly available commercial ontologies (e.g., UNSPSC (www.unspsc.org), RosettaNet (www.rosettanet.org), DMOZ (www.dmoz.org)). Currently, OWL (Web Ontology Language) formats seem to be gaining in popularity (http://www.w3.org/TR/owl-ref/).

In addition, numerous DoD specific ontology sites exists at varying classification levels. For example, a knowledge base of strategic Command & Control may already exist. If we can import this knowledge base and the ontology on which it is based, we will have not only the classification of a strategic Command & Control domain but also the first pass at the classification of the characteristics used to distinguish and describe the Command & Control functions and processes

For this exercise however we will assume that no relevant ontologies already exist and start developing the ontology from scratch.

Step 3. Enumerate important terms in the ontology

It is useful to write down a list of all the words or expression we would like either to make statements about or to explain to a user. What are the terms we would like to talk about? What properties do those words possess? What would we like to say about this vocabulary? In general, *nouns* (entities / things) become the objects within our ontology, *verbs* tend to materialize as relationships and *adjectives* and *adverbs* become the descriptive attributes that modify objects and relationships. So, the first step in actually constructing the ontology is to list all of the relevant nouns, verbs, adjective and adverbs we can think of and start linking them together. For example, the two most important Command & Control terms would likely be command and control, with control probably breaking down to operational control and technical control.

- *Command* is the authority that a commander exercises over his subordinates by virtue of his rank or assignment. Command includes the authority and responsibility for effectively using available resources and for planning, organizing, directing, coordinating, and controlling military forces for the accomplishment of assigned missions. It includes responsibility for health, welfare, training, and discipline of assigned and attached personnel.

- *Operational Control* is that control which comprises functions of command involving composition of subordinate forces, assignment of tasks, designation of objectives, and the authoritative direction to accomplish the mission. Operational control is delegated by authority of the individual who has overall force control. It does not include administration, discipline, and internal organization or unit training.

- *Technical Control* is defined as that specialized or professional guidance exercised by an authority in technical matters.

Initially, it is important to get a comprehensive list of terms without worrying about overlap between the concepts they represent, relations among the terms, nor any properties that the concepts may have, or whether the concepts are objects or attributes. A few terms we could consider might include Agile C2 attributes [JC2FC, 2003], Network-Centric Warfare tenets, attributes, and levels [NCOW-RM, 2003] or, as suggested by Haglich, *et al* [Haglich, 2000]:

Situation Awareness
Course of Action (COA)
Operational Concept
Phase COA
Recommended COA
COA Analysis
COA Military End State
COA Conflict Termination Condition
COA Status
Commander’s Estimate
Constraint/Restraint
Effect Action
Object of Effect
Target
End State Condition

End State Military Condition
End State Political Condition
Measure of Merit
Binary Measure
Numerical Measure
Objective
Desired Effect
National Security Objective
Operational Air Objective
Tactical Air Objective
Task
Airlift Task
Defensive Task
Essential Implied Task
Essential Specified Task

Force Application Task
Ground Support Task
ISR Task
IW Task
Implied Task
Jammer Task
Specified Task
Tactical Air Task
Resource to Task Assignment
Sortie
Strategic Situation Analysis
Military Unit

The next two steps—developing the class hierarchy and defining properties (attributes) of concepts—are closely intertwined. It is hard to do one of them first and then do the other. Typically, we create a few definitions of the concepts in the hierarchy and then continue by describing properties of these concepts and so on. These two steps are also the most important steps in the ontology-design process. We will describe them here briefly.

Step 4. Define the classes and the class hierarchy

There are several possible approaches in developing a class hierarchy [Uschold, 1996].

- A top-down development process starts with the definition of the most general concepts in the Command & Control domain and subsequent specialization of the concepts. For example, we can start with creating classes for the general concepts of Command & Control. Then we specialize by creating some of its subclasses. For example, Operational Control and Technical Control. We can further categorize Operational Control into lesser subclasses such as: composition of subordinate forces, assignment of tasks, designation of objectives, the authoritative direction to accomplish the mission, and so on.

- A bottom-up development process starts with the definition of the most specific subclasses, the leaves of the hierarchy, with subsequent grouping of these subclasses into more general concepts. For example, we might start by defining the lesser subclasses as: the composition of subordinate forces, assignment of tasks, designation of objectives, and the authoritative direction to accomplish the mission. We then create a common superclass for these four lesser subclasses—Operational Control—which in turn is a subclass of Command & Control.

- A combination development process is a combination of the top-down and bottom-up approaches: We define the more salient concepts first and then generalize and specify them appropriately. We might start with a top-level concept such as Command & Control, and a few specific concepts, such as the composition of subordinate forces, assignment of tasks, designation of objectives, and the authoritative direction to accomplish the mission. We can then relate them to a middle-level concept, such as Operational Control. Then we may want to generate all of the operational and technical control processes, thereby generating a number of middle-level concepts.

None of these three methods is inherently better than any of the others. The approach depends strongly on our view of the Command & Control domain and the availability of information. If a systems engineer has a systematic top-down view of the domain, then it may be easier to use the top-down approach. The combination approach is often the easiest for many ontology developers, since the concepts "in the middle" tend to be the more descriptive concepts in the domain [Rosch, 1978].

If one tends to think of Command & Control by distinguishing the most general classification first, then the top-down approach may work better. If we would rather start by getting grounded with specific examples, the bottom-up approach may be more appropriate.

Whichever approach one chooses, we typically start by defining classes. From any list created in the above examples, we select the terms that describe objects (nouns) having independent existence rather than terms that describe these objects. These will be classes in the ontology and will become anchors in the class hierarchy. We organize the classes into a hierarchical structure by asking if, by being an instance of one class, the object will necessarily (i.e., by definition) be an instance of some other (higher level) class.

Ontologies typically adhere to basic object-oriented concepts where subclasses are considered part of superclasses and attributes are inherited down the hierarchy. If a class A is a superclass of class B, then every instance of B is also an instance of A. In other words, the class B represents a concept that is a "kind of" A.

For example, every "designation of an objective" is necessarily an element of operational control. Therefore the "designation of an objective" class is a subclass of the operational control class.

Step 5. Define the properties of classes—attributes

The classes alone will not provide enough information to answer the competency questions from Step 1. Once we have defined some of the classes, we must describe the internal structure of concepts.

We have already selected classes from the list of terms we created in Step 3. Most of the remaining terms are likely to be either properties of these classes (adjectives and adverbs) or relationships between them. For example, one should quickly understand that "composition of subordinate forces" would include, for example, all those items that would describe such a composition.

For each property in the list, we must determine which class or classes it describes. These properties become attributes attached to classes. Thus, the "composition of subordinate forces" class might have the following attributes: size, operational mission, designators, etc. In general, there are several types of object properties that can become attributes in an ontology:

- "intrinsic" properties. **Intrinsic** meaning "*situated within or belonging solely to the 'composition of subordinate forces' on which it acts.* Examples would be the designators of those officers assigned duties and attached to the various forces (platoons, squadrons, etc.).

- "extrinsic" properties. **Extrinsic** meaning "*not forming an essential part of a thing or arising or originating from the outside;* such as the current location of the forces;

- parts, since the object *'composition of subordinate forces'* is structured; these can be both physical and abstract "parts" (e.g., the composition being made up of various squadrons)

- relationships to other individuals; these are the relationships between individual members of the class and other items (e.g., the commander of a squadron)

Step 6. Define the facets of the attributes

Attributes can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features attribute values can take. For example, the value of a name attribute (as in "operational mission") is one string. That is, name is an attribute with value type String.

As attribute definition is perhaps the hardest part of ontology development (e.g., attribute cardinality, attribute-value type, domain and range of an attribute, etc.), the authors have elected not to go into much more detail in this step. However, attributes can perhaps be thought of as the "architectural atoms" that eventually make up the domain of Command & Control [Curts, 1999].

Step 7. Create instances

The last step is creating individual instances of classes in the hierarchy. Defining an individual instance or member of a class requires (1) choosing a class, (2) creating an individual instance of that class, and (3) filling in the attribute values. For example, we can create an individual instance such as 'electronic jamming' to represent a specific type of aircraft squadron. So, electronic jamming is an instance of the class squadron which represents all squadrons. This instance may have the following attributes defined:

- Aircraft Type(s):
- Aircraft Designator(s)
- Primary or Secondary Mission(s):
- Number of Aircraft:
- Squadron Strength:
- Squadron Location:
- Commanding Officer:
- Aircraft Location(s):
- Etc.

5.0 SUMMARY

Ontologies have only recently become popular and seem to be taking a major role in the building of architectures. Ontologies are becoming the *de jour* method of describing domains and their internal and external relationships. While architectures and ontologies were developed for different purposes and, in fact, are not the same, many of the constructs provide similar information. Combining architectures and ontologies appears to be gaining acceptance. In general, architectures tend to provide a physical description of the organizations and systems that make up a domain while the ontology describes the more abstract concepts. Combining the physical with the abstract is a faceted approach that should be considered within the Command & Control domain.

What the authors have attempted to present in this paper is a stepwise transition to understanding Command & Control not through its various definitions but through the abstract concepts of a Command & Control domain. The value comes as we capture how people think about things in the Command & Control domain, not its definition. There is, of course, always the need for correct and understandable terminology. In fact, the effort of building a Command & Control ontology begins there by first building a vocabulary that provides controlled terminology for the Command & Control domain. This vocabulary is then organized into a taxonomy where key concepts are identified, and finally these concepts are defined and related to create an ontology.

6.0 FURTHER DISCUSSION ON ONTOLOGICAL CONSTRUCTS AND THEIR APPLICABILITY TO DOMAINS AND ARCHITECTURES

6.1 Ontological Constructs as Building Blocks. There is a need for a common description for both ontologies and architectures. Indeed, many of the constructs are the same especially at the lower, more specific, detailed levels. As we develop an ontology from abstract concepts and begin to add specificity or depth, we eventually end up with individual functions as the leaf nodes. Architectures, on the other hand are typically developed from the functional requirements (in the case of notional architectures) or functional capabilities (in the case of physical, existing architectures) that we consider essential to warfighting. These requirements have been expressed in many forms and at varying levels of granularity. However, if we adopt the concept of *functional requirements*[1] as the building blocks of our architectural description, we have the opportunity to conduct direct comparisons of effectiveness, interoperability and a large variety of other descriptors that are of interest to us (Figure 2).

[1] The term "functional requirements" will be used henceforth to include both the functions that we wish to perform (requirements) and the functions that existing systems actually do perform (capabilities).

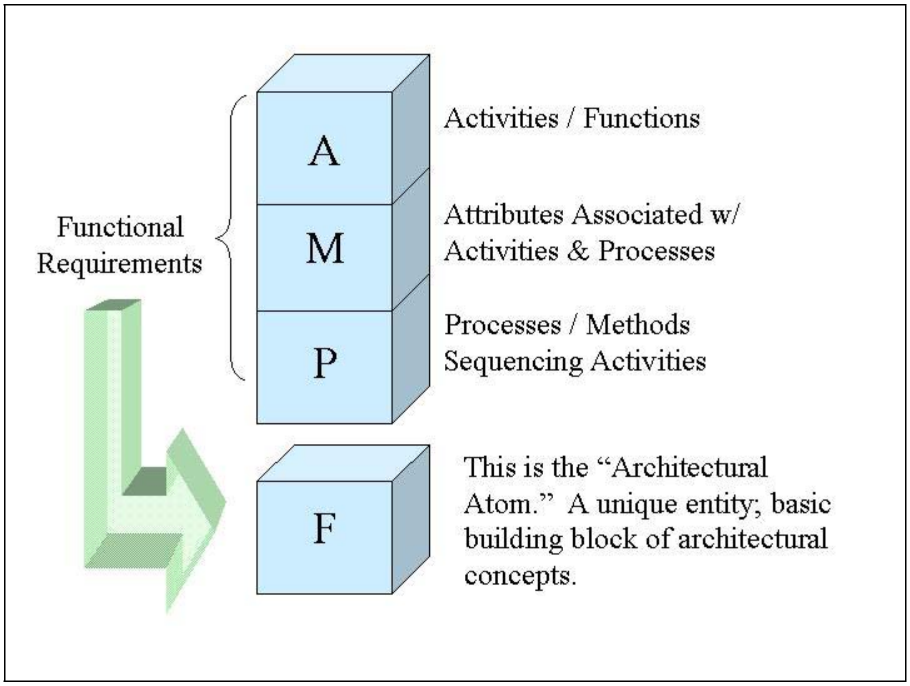

Figure 2. Basic Building Blocks.

Functional requirements lend themselves quite nicely to the object-oriented approach previously described. First, they can be represented as objects where each function or activity has a name; there are attributes associated with that function, and processes, methods or activities are performed by that function.

In this definition, Functional Requirement Objects become "*Architectural Atoms*," the building blocks from which any and all architecture components can be constructed.

Thus functions become components, components become systems and systems, in turn form a system of systems, or suite (Figure 3). At some point, as we aggregate and generalize up the hierarchy one could argue that we cross the boundary between architecture and ontology. The point here, at least in the authors' view, is that the two are very closely tied. From an architectural perspective these "Architecture Atoms" also allow us to readily identify shortfalls (gaps in our functional capabilities) and functional redundancies (overlapping capabilities from multiple suites, systems or components) for further analysis. Shortfalls usually require attention while redundancies are often intentional and required in military systems. Some redundancies, however, may be targeted for elimination in the name of efficiency and/or cost effectiveness.

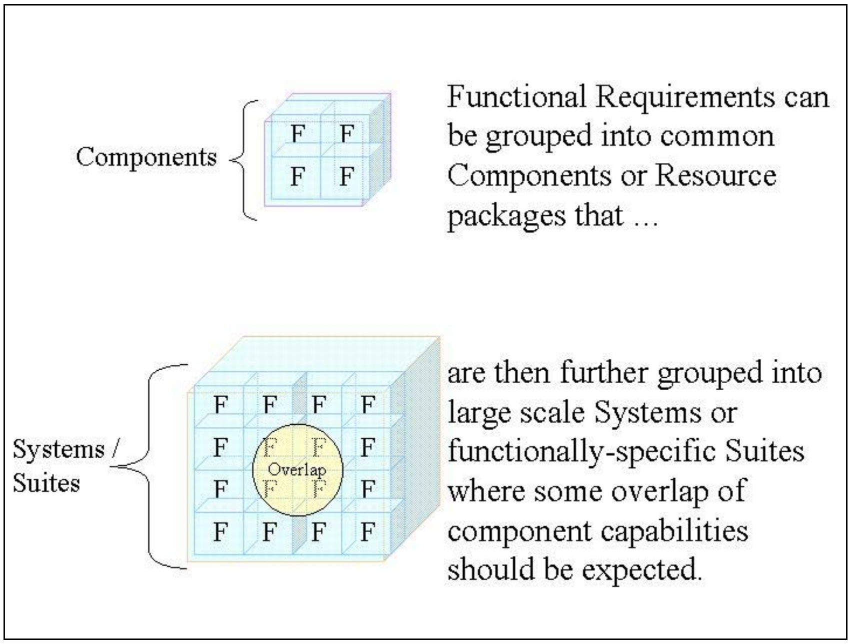

Figure 3. Functions and Components.

Thus, from a functional perspective, the entire architecture can be described using combinations of Functional Requirements (Figure 4).

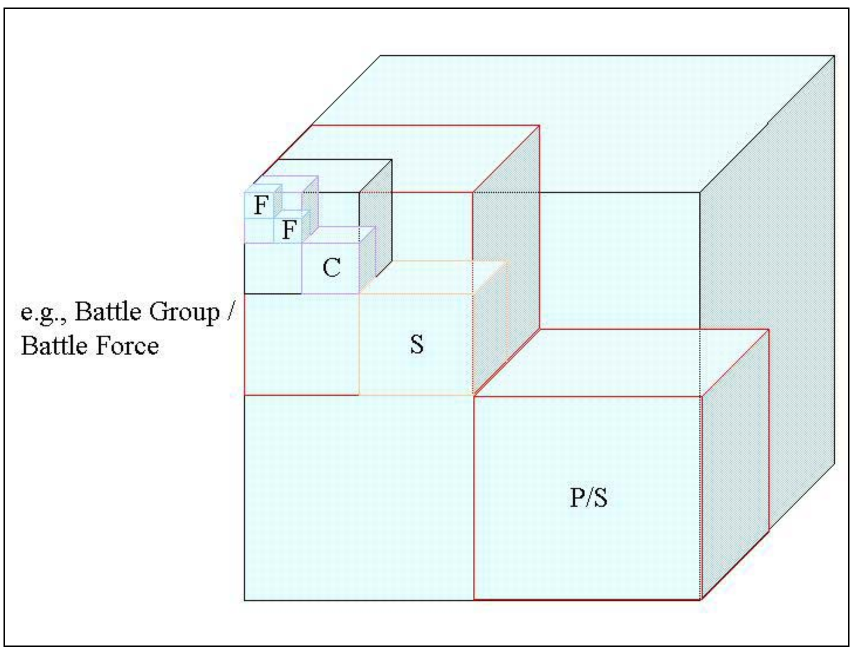

Figure 4. Building Blocks.

Object-Oriented architectural components, when assembled, might resemble a Rubik's Cube (Figure 5). Each module represents a unique unit, system, or capability that can be combined with others in a virtually limitless number of ways. In addition to this physical flexibility, once assembled, the architecture can be viewed from multiple perspectives (also nearly limitless) to satisfy the requirements of the viewer.

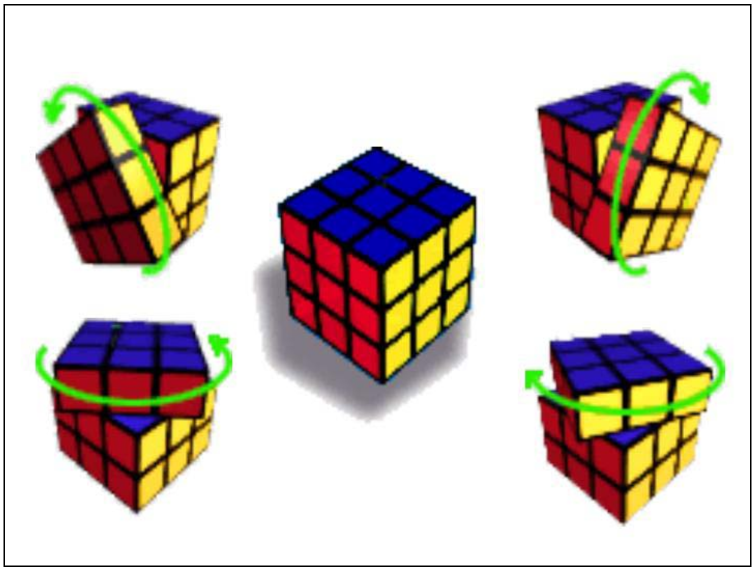

Figure 5. Rubik's Architecture Cube.

This is one of the major disappointments with architectures today and a primary reason that our systems are still not interoperable despite more than ten years identifying the issues. Numerous studies have shown that many useful architectures and architectural constructs exist. Unfortunately, they were all developed by different organizations, for different purposes, using similar but differing data, at varying levels of detail. Most were captured as documents (text and graphics) rather than as manipulable data and none, to the authors' knowledge, have been satisfactorily linked to the domain ontology. Though undoubtedly useful to the owners and developers of each, they cannot be directly combined nor compared in any meaningful way. Information Assurance (IA) has been a significant driver in Information Warfare (IW) circles recently. However, IA cannot be accomplished without interoperability, and we are not likely to achieve interoperability without a solid architectural foundation [Curts, 1999] [Curts, 2000].

Traditional database systems are limited in their data abstraction and representation power, and they fall short of providing important information management capabilities required by certain applications such as office information systems, personal databases, design engineering databases, and artificial intelligence systems.

The use of ontologies and object-oriented architecture methodologies to support the decision-maker at various levels of abstraction is an important emergent area where great strides can be made relatively quickly.

REFERENCES

[Alberts, 2003] Alberts, David S. and Richard E. Hayes. Power to the Edge. Washington, DC: U.S. Department of Defense, Command & Control Research Program (CCRP), 2003.

[Andrews, 1990] Andrews, Timothy, and Craig Harris. "Combining Language and Database Advances in an Object-Oriented Development Environment." Readings in Object-Oriented Database Systems, 186-196. Stanley B. Zdonik and David Maier, eds. San Mateo, CA: Morgan Kaufman, 1990.

[Auger, 2004] Auger, Alain. Capturing and Modeling Knowledge Objects: The SACOT (Semi-Automatic Construction of Ontologies from Texts) Project. Proceedings of the 24th Army Science Conference, Orlando, FL, 29 November – 2 December, 2004.

[Bourry-Brisset, 2000] Bourry-Brisset Anne Claire. Knowledge Modeling and Management for Command and Control Environments, Command & Control Research and Technology Symposium (CCRTS), Naval Postgraduate School, Monterey, CA, June 26-28, 2000.

[Chance, 2003] Chance, Samuel G. and Marty G. Hagenston. Assessing the Potential Value of Semantic Web Technologies in Support of Military Operations. Thesis, Naval Postgraduate School, Monterey, CA, 2003. 289 pp.

[Chaum, 2003] Chaum, Erik. Transformation Cornerstone – A Shared Ontology. Presented at the Extensible Modeling & Simulation Framework (XMSF) Workshop at the Modeling, Virtual Environments and Simulation (MOVES) Institute, Naval Postgraduate School, Monterey, CA. May 2003.

[CMC, 1995] CMC/CNO Memorandum. Information Warfare / Command and Control Warfare (IW/C2W). 16 February 1995.

[Curts, 1989a] Curts, Raymond J. A Systems Engineering Approach to Battle Force Architecture. Fairfax Station, VA: Strategic Consulting, Inc. (SCI), 1989.

[Curts, 1989b] Curts, Raymond J. An Expert System for the Assessment of Naval Force Architecture. Fairfax Station, VA: Strategic Consulting, Inc. (SCI), 1989.

[Curts, 1999] Curts, Raymond J., and Campbell, Douglas E. "Architecture: The Road to Interoperability." Paper presented at the 1999 Command & Control Research & Technology Symposium (CCRTS), U.S. Naval War College, Newport, RI, June 29 – July 1, 1999.

[Curts, 2000] Curts, Raymond J., and Campbell, Douglas E. "Naval Information Assurance Center (NIAC): An Approach Based on the Naval Aviation Safety Program Model." Paper presented at the 2000 Command & Control Research & Technology Symposium (CCRTS), U.S. Naval Postgraduate School, Monterey, CA, June 24-28, 2000.

[Curts, 2001] Curts, Raymond J. and Douglas E. Campbell. Avoiding Information Overload Through the Understanding of OODA Loops, A Cognitive Hierarchy and Object-Oriented Analysis and Design. Annapolis, MD: C4ISR Cooperative Research Program (CCRP), 2001.

[DSBIAB, 1994] Report of the Defense Science Board Task Force on Information Architecture for the Battlefield. Washington, DC: Office of the Under Secretary of Defense for Acquisition and Technology (USD(A&T)), 1994.

[Fikes, 1997] Fikes, Richard and Daphne Koller. Large-Scale Repositories of Highly Expressive Reusable Knowledge. Presented at the High-Performance Knowledge Bases (HPKB) Program West Coast Introductory Meeting (March 27, 1997) as part of the Stanford University's Knowledge Systems Laboratory, sponsored by the Defense Advanced Research Projects Agency (DARPA).

[Fowler, 1997] Fowler, Martin with Kendall Scott. UML Distilled: Applying the Standard Object Modeling Language. Reading, MA: Addison-Wesley, 1997.

[Gauvin, 2004] Gauvin, Marlene, Anne-Claire Bourry-Brisset and Alain Auger. Context, Ontology and Portfolio: Key Concepts for a Situational Awareness Knowledge Portal. Proceedings of Hawaii's International Conference on System Sciences (HICSS-37), January 5-8, 2004.

[Gouin, 2003] Gouin, Denis, Marlene Gauvin and Elizabeth Woodliffe. Towards a Situational Awareness Knowledge Portal. Proceedings of SPIE 2003 on Aerosense/Defense Sensing, Simulation and Controls, Orlando, FL, 21-25 April 2003. Volume 5101 under Battlespace Digitization and Network-Centric Systems III.

[Gruber, 1993] Gruber, T. "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition*, 5(2), pp. 199-220, 1993.

[Gruninger, 1995] Gruninger, Michael and M.S. Fox. Methodology for the Design and Evaluation of Ontologies. From the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, 1995.

[Haglich, 2000] Haglich, Pete and Gary Edwards. An Ontology for Joint Air Operations. Deliverable developed for the Air Force Aerospace Command and Control, Intelligence, Surveillance and Reconnaissance Center by ISX Corporation under Contract M06942-S003-KG12-43065, March 17, 2000.

[Hutcherson, 1994] Hutcherson, Norman B., LtCol USAF. "The Five Pillars of Command and Control Warfare," and "Command and Control Warfare—As A War-Fighter's Tool." *Command and Control: Putting Another Tool in the War-Fighter's Data Base*, a research report for Air University Press, Maxwell Air Force Base, Alabama, 1994.

[IWIP, 1996] Curts, Raymond J. and Charles Ristorcelli. Information Warfare Program Directorate (PD-16). Information Warfare Implementation Plan and Acquisition Strategy. Washington, DC: Space & Naval Warfare Systems Command (SPAWAR), 1996.

[JC2FC, 2003] Chairman, Joint Chiefs of Staff (CJCS – J6). Joint C2 Functional Concept: Agile C2, Version 5.0. Washington, DC: U.S. Joint Chiefs of Staff, 7 May 2003.

[JP 1-02, 1994] Joint Chiefs of Staff (JCS) Publication 1-02. The DoD Dictionary of Military and Associated Terms. Washington, DC: U.S. Joint Chiefs of Staff.

[JP 3-13, 1987] Joint Publication 3-13.1, *Joint Doctrine for Command and Control Warfare (C2W).*

[MOP 30] Chairman, Joint Chiefs of Staff (CJCS) Memorandum of Policy (MOP) #30. Command and Control Warfare (C2W). Washington, DC: U.S. Joint Chiefs of Staff, 8 March 1993.

[NCOW-RM, 2003] Assistant Secretary of Defense, Networks and Information Integration (ASD(NII)). GiG Net-Centric Operations and Warfare Reference Model. Washington, DC: U.S. Department of Defense (DoD), December 2003.

[NDP 6] Naval Doctrine Publication #6. Naval Command and Control. Washington, DC: U.S. Navy, 1 September 1994.

[NISL Pub 1] Naval Information System Library, Publication #1. The Rise of Modern Command and Control. Washington, DC: The Renaissance Group of Washington, DC, 1995.

[Noy, 2001] Noy, Natalya F. and Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Unpublished Technical Report SMI-2001-0880, Stanford Medical Informatics and KSL-01-05, Knowledge Systems Lab, Stanford University, Stanford, CA, 94305. Available at http://protege.stanford.edu/publications/ontology_development/ontology101.html

[Percival, 1995] G.S. Percival. Application of Complexity Theories to Evolutionary System Development. Paper presented at the Evolution of Complexity Symposium, May 29 / June 3, 1995, at the Free University of Brussels, Belgium. gperciva@eos.hitc.com

[Prieto-Diaz, 2002] Prieto-Diaz, Ruben. A Faceted Approach to Building Ontologies. Unpublished paper from the Commonwealth Information Security Center, James Madison University, VA, dated February 2002, 18pp. prietodiaz@cisat.jmu.edu.

[RMC, 1995] Report of the Commission on Roles and Missions of the Armed Forces. Directions for Defense. Arlington, VA: U.S. Department of Defense (DoD), 1995.

[Rosch, 1978] E. Rosch, E. Principles of Categorization. *Cognition and Categorization*. R. E. and B. B. Lloyd, editors. Hillside, NJ, Lawrence Erlbaum Publishers**:** 27-48, 1978.

[Uschold, 1996] Uschold, M., and Michael Gruninger. *Ontologies: Principles, Methods, and Applications.* Knowledge Engineering Review 11(2): 93-155, 1996.

[Zaniolo, 1986] Zaniolo, Ait-Kaci, Beech, Cammarata, Kerschberg and Maier. "Object Oriented Database Systems and Knowledge Systems." Expert Database Systems: Proceedings from the 1st International Workshop. Larry Kerschberg, ed. Menlo Park, CA: Benjamin Cummings, 1986.

