

**2006 CCRTS
THE STATE OF THE ART
AND THE STATE OF THE PRACTICE**

Title:

**The Role of Ontology
in System-of-Systems Acquisition**

Relevant Topics:

C2 Architecture
C2 Modeling and Simulation
C2 Concepts and Organizations
C2 Analysis
Network-Centric Metrics
Policy

Authors:

Israel Mayk, D.Eng.Sc.
Battle Command Division
US Army RDECOM CERDEC C2D
Fort Monmouth, NJ, USA

Azad M. Madni, Ph.D.
CEO and Chief Scientist
Intelligent Systems Technology, Inc.
Santa Monica, CA, USA

Point of Contact:

Dr. Azad Madni
Intelligent Systems Technology, Inc.
3250 Ocean Park Blvd., Suite 100
Santa Monica, CA 90405
310-581-5440 x144
fax: 310-518-5430
amadni@intelsystech.com

The Role of Ontology in System-of-Systems Acquisition

Israel Mayk, D.Eng.Sc.
Battle Command Division
US Army RDECOM CERDEC C2D
Fort Monmouth, NJ, USA
Israel.mayk@us.army.mil

Azad M. Madni, Ph.D.
CEO and Chief Scientist
Intelligent Systems Technology, Inc.
Santa Monica, CA, USA
amadni@intelsystech.com

***Abstract** - This paper addresses the importance of a unified ontology for a Battle Command (BC) system of systems (SoS) acquisition. A BC SoS is a Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance and Target Acquisition (C4ISR&TA) federation of large-scale, net-centric systems that are collaborative and interoperable and include heterogeneous multi-agency managed intelligent agents, humans-in-the-loop, and unmanned autonomous systems. As systems become increasingly complex, modularity becomes the key to reuse, scalability, and an open architecture. In addition, these design features are key to a manageable and affordable transformation from current to future capabilities across acquisition maturity phases over several decades of fielding. A new large-scale SoS cannot be built in isolation. It needs to evolve internally and accommodate external pressures to integrate or be interoperable with current “systems of record.” The development of a unifying ontology that spans multiple domains in the SoS is shown to be crucial, if not pivotal, to the success of SoS engineering efforts which are inherently multi-disciplinary and collaborative.*

Keywords: Ontology, Reference Model, Markup Language, Architecture, Information, System, Enterprise, Acquisition, Research, Development, Engineering, Reuse.

1. Introduction

The importance of a unified ontology to facilitate and guide the acquisition process cannot be overemphasized [1, 2], especially for the acquisition of complex system of systems (SoS). SoSs are large-scale, netcentric enterprises that contain a mix of multi-agency, heterogeneous elements including intelligent agents, humans-in-the-loop subsystems, and unmanned autonomous components. Examples of complex SoSs in the US Army are the Army Battle Command System of the Current Force [3] and the Future Combat System of the Future Force [4]. Typical systems that would be found in a Joint Service (Army, Navy, Marine and Air Force) SoS are depicted in Figure 1. In general, the acquisition of SoSs is an ongoing process since at any point in time there is always a “current force” that the acquisition enterprise is engaged in improving, while it concurrently develops the next generation “future force.”

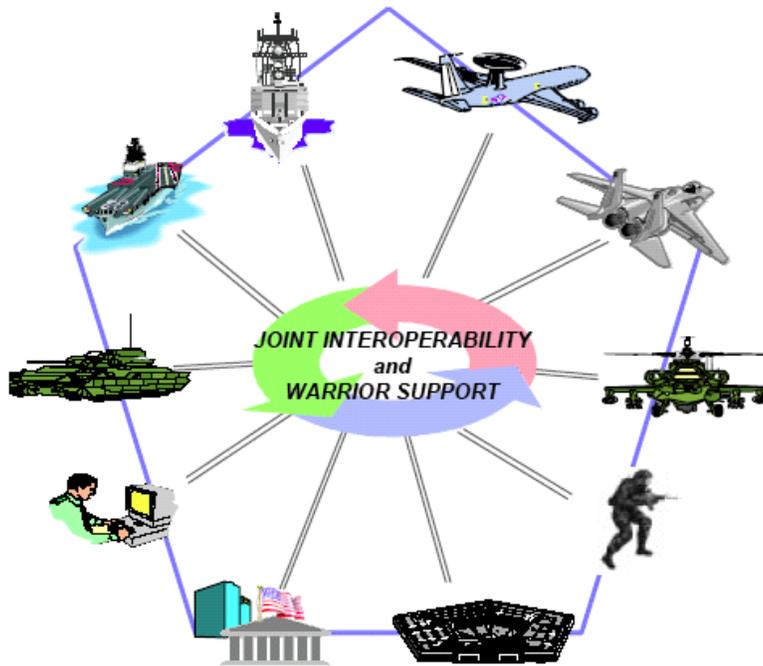


Figure 1. Battle Command C⁴ISR&TA SoS spans Multinational to Joint Tactical Echelon Systems [5]

In its simplest form, an ontology can be thought of as a type of graph-based representation that defines two key primitives: node (which represents concepts) and arc (which represents node-to-node relationship). Specifically, the nodes represent conceptual elements in a given domain, while the arcs represent the relationships between and among the conceptual elements. A dictionary, comprising both types of concepts, is an essential part of the ontology. In addition, an ontology should provide both an information model and a reference model as a basis for the language that is used to represent and express complex information constructs and application products. The *information model* includes an information architecture and associated axioms and rules for semantic interoperability based on a data model. The *data model*, in turn, should include a taxonomy and associated markup language or meta-data tagging of the data elements along with the specification of allowable data values. ISO/IEC 11179 is a standard that provides a convenient baseline for specifying and standardizing data elements [6]. While it provides meaningful guidance for generating a taxonomy, it does not address the broader issues of defining an ontology.

All too often, acquisition managers, engineers and scientists emphasize the physical (Personnel and Materiel) aspects of a system while neglecting the informational aspects that are critical to successful system acquisition. The DoD Architectural Framework (DoDAF) [7] identifies up to 26 views or templates for describing a system, but only one view, OV7, is devoted to the logical data model of a system. Furthermore, DoDAF does not advocate an overarching information model to apply to all C⁴ISR&TA applications in a uniform manner. While referencing the voluminous C⁴ISR Architecture Data Model (CADM), DoDAF does not provide a reference model or guidance for the potential use

of CADM to represent the content associated with DoDAF views. This missing ingredient is referred to herein as the **Unified Ontology**.

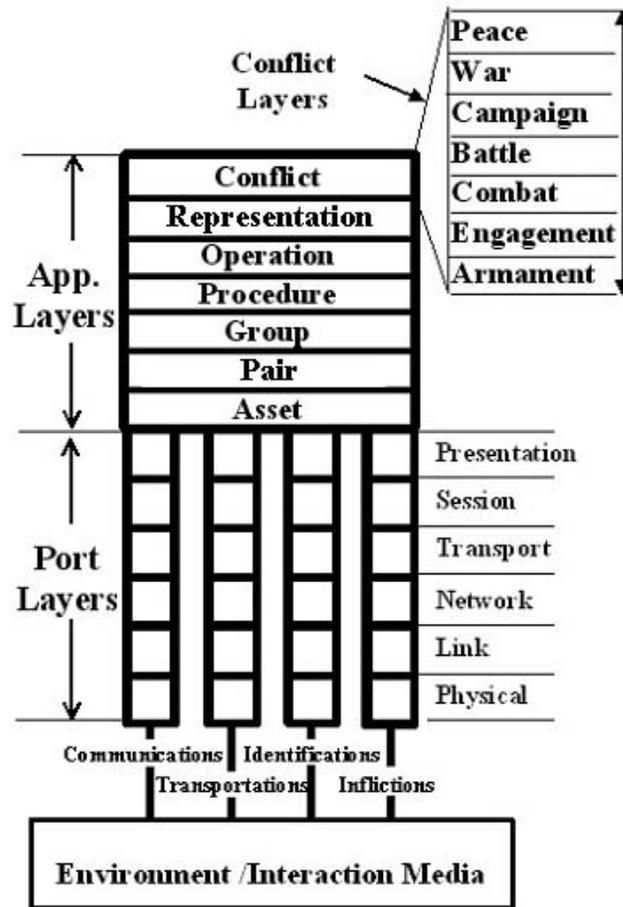


Figure 2. The C2RM Enterprise/Unit Domain View [10]

The Unified Ontology typically includes: a) a generic overarching BC information model (IM) [8] expressed by a set of comprehensive, and synergistic markup languages (XML*) represented in XML [9], whose instances could be validated by the XML schema; and b) a corresponding reference model (RM), such as the C2RM [10] depicted in Figure 2, to represent not only the DoDAF views but also the more technical Model-Driven Architecture (MDA) [11] views represented, for example, in UML* (MOF [12], UML [13], and SysML [14]). The foregoing description leads to the following comprehensive definition of a unified ontology:

$$\text{Unified Ontology} = \{ \text{IM (XML}^*) + \text{RM(UML}^*) \}$$

2. Acquisition and System Engineering

While much has been written on acquisition and systems engineering, to date there are no formal ontologies for either of these domains. The actual creation of an ontology for the acquisition domain or for the system engineering domain is beyond the scope of this paper. Nevertheless, one can acquire a great deal of insight into what constitutes an ontology for these two domains by analyzing the use of keywords in documents relevant to each domain. One can assess the mutual proximity of

the two domains by analyzing the frequency of occurrence of keywords and assessing the extent of the similarity and the opportunity to share a common upper ontology layer based on the relative frequency of overlapping concepts. To illustrate this point, let us consider the DoD Manager’s Guide to Technology Transition in an Evolutionary Acquisition Environment [15] as an instance of a key document addressing the acquisition domain, and the IEEE Standard for Application and Management of the Systems Engineering Process [16] as a sample document representing the system engineering domain. A list of approximately 15-20 keywords was independently selected for each document. After removing duplicate keywords from each list, a common list was formed and the frequency of occurrence was extracted. The integrated results are shown in Table 1. One immediate observation that can be made is that the two domains have much in common, and therefore, can clearly benefit from a common ontology. In addition, this type of analysis can be applied to additional domains under the acquisition process such as those covered by the operational and technical views of DoDAF.

Table 1. Acquisition Ontology Keyword Occurrence within DoD Manager’s Guide to Technology Transition [15] and IEEE Standard 1220-1988 [16]

RANK		KEYWORD	OCCURS	
[15]	[16]		[15]	[16]
1	16	Techn(ology)	1679	214
2	1	System	631	886
3	10	Develop(ment)	500	212
4	25	Acqui(sition)	426	2
5	2	Require(ment)	372	451
6	14	Financ(ial)/Cost	314	96
7	24	Government	299	4
8	26	Research	235	2
9	15	Inform(ation)	232	58
10	4	Product	227	378
11	22	Industry	192	5
12	8	Process	187	270
13	13	Test	125	123
14	3	Design	87	382
15	11	Data	84	140
16	27	Science	84	0
17	18	Evaluat(ion)	81	35
18	7	Engineer	80	271
19	9	Life(cycle)	74	235
20	19	Know(ledge)	69	16
21	21	Secur(ity)	46	6
22	6	Function	43	288
23	17	Model	28	38
24	20	Simulat(ion)	23	13
25	12	Architect(ure)	13	131
26	28	Academia	7	0
27	23	Reus(e)	4	5
28	5	Enterprise	3	361

2.1. The Acquisition Enterprise

In the acquisition domain, a SoS may be viewed as an enterprise within the acquisition enterprise. Therefore, a truly foundational ontology for the acquisition domain should be applicable and usable for the SoS domain. Like many related terms such as shown in Table 1, “acquisition” can be used in a restrictive sense to mean solely the purchase of a finished product, i.e., a fully developed SoS in the form of a single “out-of-the-box, ready-to-use” product. Due to cost, complexity and security, in the C4ISR&TA domain, one simply cannot “shop” for a SoS. The broadest definition of “acquisition” would include all activities from “cradle to grave” that are needed to sustain a SoS. These include concept and technology definitions of the enterprise problem domain in terms of concepts and technologies needed for Future Operational Capabilities (FOCs), research, development, engineering, production, fielding, operation, maintenance, post-deployment support, replacement and disposal. It is important to realize that the SoS problem domain is a sub-domain of the acquisition enterprise problem domain and, as such, solutions for the acquisition enterprise would apply to the SoS problem. C4ISR&TA SoSs are so complex that the acquisition cycle needs to be applied iteratively in what is known as *spiral blocks* [15]. We define the set of organizations responsible for the acquisition of a SoS as the Acquisition Enterprise. Unlike the IEEE Standard 1220 [16] definition of an enterprise which is limited to the engineering organization this is an all-inclusive definition. As such, the Acquisition Enterprise is also very complex, consisting of a number of cooperating and, at the same time, competing Communities of Interest (CoIs), stakeholders, advocates and proponents. These entities can be characterized in a two-dimensional space, with one axis corresponding to institutional affiliations (e.g., multi-national governments, industry and academia) and the second axis representing the various roles that come into play in the acquisition enterprise. The latter includes operational users, doctrinal developers, trainers, testers, evaluators, technology developers, system developers, researchers, and various sub-domains Subject Matter Experts (SMEs).

The difficulty in defining an Acquisition Enterprise is exemplified in the DoD Manager’s Guide to Technology Transition in an Evolutionary Acquisition Environment [15], which defines “acquisition” as:

“the act of acquiring goods or services for directly benefiting the government or for its use, e.g., buying something that the government needs.”

It is interesting to note that the above definition is circular in that the term “acquire” in the definition is simply the verb form of “acquisition.” Consequently, no added insight is provided relative to what it should entail or what it should include. The term “acquire” is also used to define Best Value as:

“represented by an item or process that consistently performs the required function and has the lowest total cost. Best value includes increased performance as well as reduced costs for developing, producing, acquiring, and operating a system.”

The above definition indicates that Best Value needs to be optimized over the lifecycle of a system. At the same time, it seems to imply that development, production, and operations of a SoS are not included in the acquisition lifecycle. This is probably not the case but it does illustrate how

standalone definitions can result in ambiguity and even give rise to contradictions which, in a complex domain, can only be resolved by developing and invoking a formal ontology of the domain. Additional insights about acquisition can be extracted from the DoD Manager's Guide which defines the "acquisition community" as:

"...the program managers, product managers, staffs, and organizations that manage the development, procurement, production, and fielding of systems. They provide new, improved, or continuing materiel, weapons systems, or information system capabilities or services for a validated operational or business need."

As systems become increasingly more complex, modularity becomes key to reuse, scalability, and adherence to the tenets of an open architecture. In addition, these design features are key to a manageable and affordable transformation from current to future capabilities across acquisition maturity phases (Research, Development and Engineering) which span technology readiness levels (TRL) 1-9 [17] and extend into several decades of fielding. A new large-scale SoS cannot be built in isolation. It must evolve internally and accommodate external requirements for integration or interoperability with legacy or current "systems of record." Considering the inherently multi-disciplinary and collaborative nature of SoS transformation efforts, the recognition of the need and the investment in developing a unifying layered ontology across the acquisition domain and subdomains are key to smooth integration and interoperability, and operationally seamless synchronization, self-organization, federation, and collaboration.

In today's acquisition era, the desirability of an overarching ontology-driven environment comprising software tools and frameworks and key to realizing the benefits of a Model-Driven Architecture (MDA) is a given. Analogous to the Meta Object Facility (MOF), which provides a higher level of abstraction from which to derive UML, a reference model is needed to systematically derive the various domain ontologies, which, in turn, lead to Model-Driven Ontologies (MDO), which facilitate the generation of MDAs. A unified MDO is envisioned as a mechanism for driving, leveraging, and reusing similarities of patterns, as well as structural and behavioral paradigms across concept, design and implementation languages and across the diverse domains associated with the component subsystems of the SoS. The adoption of this approach for developing a MDO to drive MDA development can be expected to prove invaluable in achieving SoS acquisition affordability and longevity objectives.

2.2. IEEE Standard 1220

The IEEE Standard 1220 provides a semi-formal framework for describing the System Engineering Process (SEP) for information systems containing humans, hardware and software. It is important to note that the concept "information system" has two terms: "information" and "system." By extension, one can reasonably assume that an Information Engineering Process (IEP) corresponding to the SEP should also be established and, ultimately, integrated into a single, coherent, Unified Engineering Process (UEP). The SEP recognizes the need to maintain an integrated database or

repository of all information pertinent to systems engineering, to include all data, schema, models, tools, technical management decisions, process analysis information, requirement changes, process and product metrics, and tradeoffs. In a similar vein, an IEP is required to develop information products. At a minimum, the SEP should be used to define a common information model or, better yet, a comprehensive unified ontology to guide the structure of such a repository. The overall benefit and Return-on-Investment (ROI) of this strategy is the ability to “jump-start” new system acquisitions with minimal duplication of effort through maximum reuse across systems and information products. Currently, there are several ongoing efforts to standardize ontologies at multiple levels; however, none of these initiatives is concerned with establishing an IEP as a companion to the SEP. In our opinion, an IEP should be the mechanism for establishing and invoking ontology standards. The ultimate transformation of information SoS, to achieve autonomous self-organizing connectivity, federation, collaboration and semantic interoperability for net-centric operations (NCO), requires that information-product-oriented ontologies should be unified with physical-product-oriented ontologies.

2.3. A Dual Use Product Ontology

According to the IEEE dictionary, the concept of architecture provides the structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time. This definition should apply equally to both physical systems and information products. To illustrate the importance of ontology as a more complete framework for achieving semantic interoperability between the two domains, let us consider the concept of a “product.” In systems engineering parlance [10], a product may be any element or sub-element of the following product tree:

System → Assembly → Component → Part

In information engineering parlance, a product may be any element or sub-element of the following product tree:

Package → Topic → Component → Fact

Both these product trees are analogous to each other. The System Ontology is defined for physical artifacts, whereas the Package Ontology is defined for information artifacts. A system comprises elements that may include hardware, software and humans. Each element, starting with the system as the highest level element, may be thought of as a container of sub-elements at a particular level of the ontological architecture, i.e., information model. Clearly, an assembly is an element of, or is contained within, the system. But technically it would not be considered a subsystem because it is expected to have different characteristics. For example, a subsystem may contain hardware, software and humans, whereas an assembly may be limited to hardware and software. Going down the hierarchy, a component may be characterized as either hardware or software, but not both. Finally, a part is a black box that consists of an irreducible element of the system which may be a human

operator, a software algorithm or a hardware large-scale integrated circuit. Thus, the concept of “element” is reusable in a nested sense, i.e., since an “element” contains a “sub-element,” a “container” would contain a “sub-container.” Any of the above hierarchical concepts may be decomposed into sub-elements such as subsystems, subassemblies, and subcomponents for the system; and sub-packages, subtopics, and subcomponents for the information package that defines the product to be developed by an enterprise. In keeping with this terminology, DoDAF views are also considered information products!

3. Markup Languages

Markup languages play a key role in representing data, information, knowledge, and experience. Markup Languages need both syntax and semantics to be considered complete. Syntax corresponds to the grammar enforced in the information product. Semantic provides the meaning within the context of a specific instance of the information product. Full semantic interpretation of an information instance can only be achieved within the context of the domain ontology. Layered markup languages are needed to bridge the gap between an XML “character string” and an XML encoded “fact” that supports logic or cognitive reasoning. XML, the most well known syntax component for a markup language, is text-oriented and provides the underlying foundation for several other current and newly emerging markup languages [18]. RDF, DAML, and OWL can potentially enrich an XML character string with semantics needed for netcentric operations (NCOs), especially when using the web. Furthermore, these markup languages are domain-independent and, as such, require additional markup such as the one being proposed by upper ontology proponents [19]. MOF, UML and SysML are considered modeling languages which add graphic components to a markup language and embed semantics relevant to object-oriented applications more directly. Figure 3 shows a C2RM-derived UML information model, while Figure 4 presents a fragment of a C2RM-derived XML information product schema.

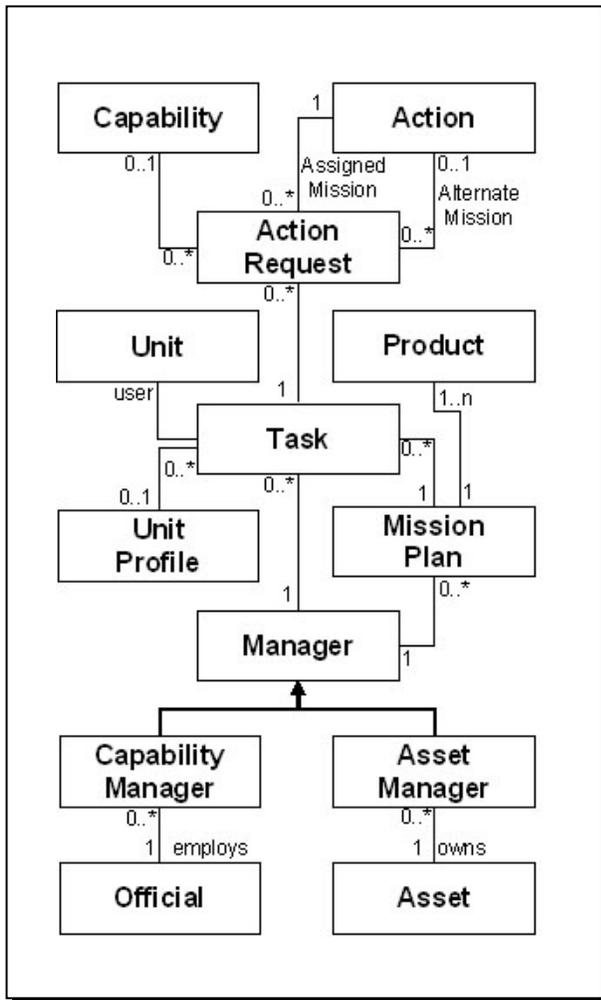


Figure 3. An Information Model for Key Enterprise Planning Concepts

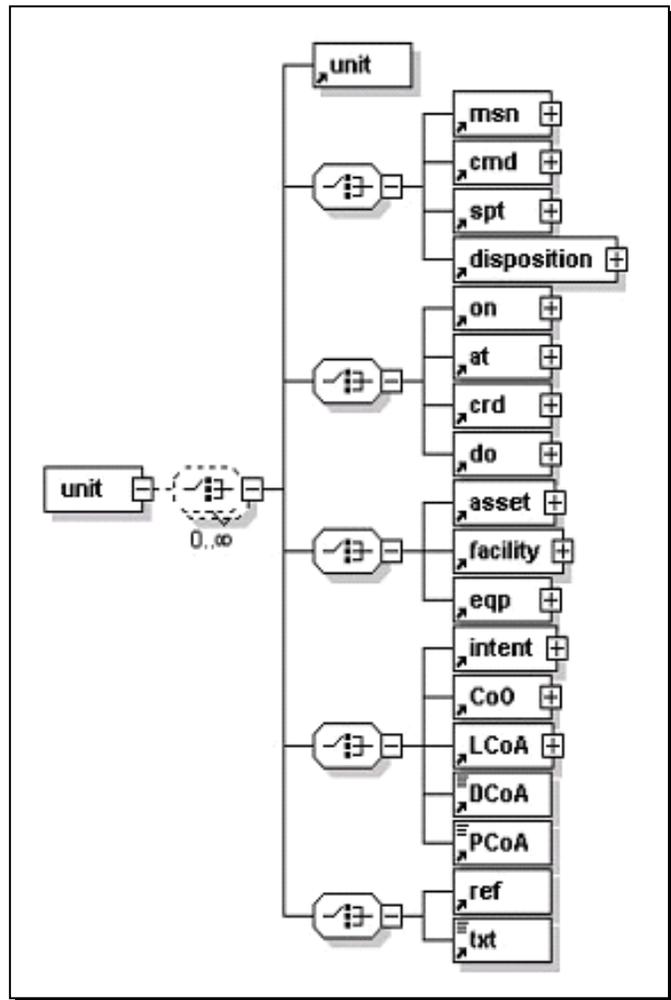


Figure 4. AN XML Schema excerpt for an Enterprise Unit Represented in a C2 Product IAW the C2RM

4. Reference Models (RM)

While the idea of a reference model is certainly not new, there has been a recent surge of interest in this area evidenced by the emergence of new reference models. Any generic model that comes with specific examples is considered a reference model. Reference models began appearing in the 1980's following the success of the ISO OSI seven-layer RM [20] that revolutionized the way communications systems are developed. Another well known RM is the Department of Defense (DoD) four-layer Technical RM (TRM) [5] which integrates and supersedes three other RMs by: a) adopting the structure of the Portable Operating System Interface (POSIX) Open System Environment (OSE) RM [21]; b) adding the service view of the Technical Architecture Framework for Information Management (TAFIM) TRM [22]; and c) incorporating the interface view of the Generic Open Architecture (GOA) RM [23] into a single uniform, tailorable RM. These RMs, however, do not address important interaction protocol issues. Furthermore, they are less modular,

and therefore, do not offer the same level of insight to the Battle Command (CoIs) that the ISO OSI RM provides to the communications CoI.

The Open Distributed Processing-Reference Model (ODP-RM) [24] is another key RM that complements the TRM. It describes systems that support heterogeneous distributed processing both within and between organizations through the use of a common interaction model. The goal of ODP-RM is to achieve portability of applications across heterogeneous platforms. The ODP-RM includes five viewpoints: Enterprise, Information, Computational, Engineering and Technology, which span the 26 DoDAF views. However, the ODP-RM also does not provide any C4ISR understanding or content to be useful in domain-oriented applications such as combat, combat support and combat service support.

On the other hand, the C2RM (Figure 2) leverages the ISO OSI RM which deals strictly with communications and extends it [10] to include all possible types of interactions inherent in BC systems that are stacked on top of primitives such as “moving” “shooting” “seeing” and “communicating.” The C2RM is a nested enterprise model that can model both the acquisition enterprise as well as the SoS and subordinates systems in a reusable way. It has been represented in UML/XML to various degrees in a number of projects. It is compatible with the ODP-RM and the DoD TRM and, with some additional effort, can also be used to support all the DoDAF views using standard semantics. It is multi-dimensional in supporting the technology views as well as the systems and operational views. It is organized along four key dimensions for characterizing domain-specific applications: Organization, Technology, Product, and User. The Organization dimension involves viewing application capabilities to manage, supervise and execute. The Technology dimension involves viewing applications capabilities to collect, store, process, present and disseminate products. The Product dimension involves application capability to handle data, information, knowledge and experience. Finally, the User dimension involves application capabilities to synchronize, federate, collaborate and operate in the mission space.

5. Conclusions and Recommendations

This paper has presented the concept of a unified ontology as a key driver of the acquisition lifecycle of current and future SoSs (e.g., BC C4ISR&TA) developed for netcentric operations. For these applications, an effective ontology is one that is layered and has nested schemas based upon markup languages and reference models grounded in synergistic combinations of XML and UML/SysML. Within this overall framework, the schemas and domain-specific reference models should be based on a comprehensive, multi-dimensional, enterprise-level reference model such as the C2 Reference Model [10].

The upper ontology layer should span the acquisition lifecycle as well as the role of the units containing the SoS and its subordinate systems in addressing a future operational capability required by the Enterprise. One can expect that as the ontology matures, so will the SoS. Scalability, stability, and reusability will be inherent in the overall architecture so that product improvements will be much

easier to undertake, and products will be much easier to adapt than is possible today. To achieve comprehensive coverage, a unified ontology must be derived from an overarching enterprise-level reference model. Such a unified ontology is key to representing the integrated mission area architecture that is used to: synchronize and manage the development of joint warfighting capabilities; assess the needs for improving current systems; and justify new systems. The unified ontology is an essential next significant step forward in pursuit of proactively synchronizing future requirements, maturing technologies, developing systems, and integrating SoSs. This paper is intended to shed light on the importance of ontologies in SoS acquisition and as a necessary first step to achieving widescale institutionalization.

6. References

- [1] Madni, A.M., Lin, W., and Madni, C.C. "IDEON™: An Extensible Ontology for Designing, Integrating, and Managing Collaborative Distributed Enterprises" in Systems Engineering: The Journal of the International Council on Systems Engineering, Vol. 4, No. 1, 2001.
- [2] Madni, A.M., Madni, C.C., and Lin, W. "IDEON™/IPPD: An Ontology for Systems Engineering Process Design and Management," (Invited Paper) Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, October 11-14, 1998, pp. 2597-2602.
- [3] Army Battle Command Systems (ABCS)
<http://www.fas.org/man/dod-101/sys/land/abcs.htm>.
- [4] Future Combat System (FCS)
<http://www.fas.org/man/dod-101/sys/land/fcs.htm>.
- [5] Department of Defense (DoD), Version 2.0 Technical Reference Model, 9 April 2001.
- [6] ISO/IEC 11179 – Information Technology
<http://metadata-standards.org/11179/>
- [7] DoD Architecture Framework: Volume I/II: Version 1.0, DoD Architecture Framework Working Group.
- [8] I. Mayk, B. Goren "C2 Product-Centric Approach to Transforming Current C4ISR Information Architectures," ICCR&T 2004, San Diego, CA.
- [9] T. Bray et al., Extensible Markup Language (XML): World-Wide Web Consortium,
<http://www.w3.org/XML/>.
- [10] 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.
- [11] MDA, <http://www.omg.org/mda/>
- [12] MOF, <http://www.omg.org/technology/documents/formal/mof.htm>.
- [13] UML, <http://www.omg.org/technology/uml/index.htm>.
- [14] SysML Specification v. 0.9 Draft (10 Jan 2005), <http://www.sysml.org/artifacts.htm>

- [15] Manager's Guide to Technology Transition in an Evolutionary Acquisition Environment, Version 1.0, January 31, 2003, Defense Procurement and Acquisition Policy, Office of the Under Secretary of Defense (Acquisition, Technology and Logistics).
- [16] IEEE Std 1220-1998, IEEE Standard for Application and Management of the Systems Engineering Process, IEEE, ©1999.
- [17] TRL, <http://www.globalsecurity.org/military/intro/trl.htm>.
- [18] Zap Think, LCC, "Key XML Specifications and Standards," Document ID: ZTS-G1101, © 2002.
- [19] John Bateman's Ontology Portal <http://www.fb10.uni-bremen.de/anglistik/langpro/webospace/jb/info-pages/ontology/ontology-root.htm>.
- [20] ISO 7498(Draft) Information Processing Systems - OSI Basic Reference Model, 1984. Also see Uyless, B, ISO OSI a Model for Communications, Prentice Hall, 1991, and Jain, B. N., and Agrawala, A. K., OSI, Its Architecture and Protocols, McGraw-Hill, NY, ©1993.
- [21] IEEE Guide to the POSIX Open System Environment (OSE) (IEEE 1003.0-1995), Institute of Electrical and Electronics Engineers, Inc., ©1995.
- [22] DoD Technical Architecture Framework for Information Management, Version 3.0. April 1996.
- [23] General Open Architecture (GOA) standard, SAE AS 4893.
- [24] ODP-RM, ISO/IEC 10746, OMG, 1995-98 <http://lamswww.epfl.ch/reference/rm-odp>.

Dr. Israel Mayk. Dr. Mayk is a Research Scientist and Technical Manager within the Command and Control Directorate, US Army Research, Development and Engineering Command (RDECOM), Communications-Electronics Research, Development and Engineering Center (CERDEC) at Fort Monmouth NJ. He is responsible for research and development of battle command knowledge-based decision support systems demonstrations and prototypes as well as for technology integration and architectures. He is currently the Chair of the Intelligent Agents sub-IPT of the RDECOM Network IPT, the Technical Manager of several Exploratory Development Programs and of the Simulation and Information Systems Connectivity Experimentation (SINCE) Program. Dr. Mayk's R&D efforts support several key Army/CERDEC C2 Programs including FCS, ABCS, DCGS-A and NCES. From 1985 to 1995 Dr. Mayk was a member of the Basic Research Group (BRG) of the C³ Research and Technology Program sponsored by the Joint Directors of Laboratories (JDL), Technical Panel for C³ (TPC3) and chaired the C² Reference Model (C2RM) Subgroup. He was recently nominated and selected for the 2005 Outstanding AMC Personnel of the Year Award. His research interests include C² decision-aiding, C² architecture modeling, systems interoperability, ontology-based integration, and agent-based systems. Dr. Mayk is a Senior Member of the IEEE and a member of AFCEA, AUSA, and USNI. He holds a B.A. degree in physics (1970) from Rutgers University, NJ, a M.Sc. degree in nuclear physics (1973) from the Weizmann Institute of Science, Israel, and an Eng.Sc.D. in electrical engineering (1985) from NJ Institute of Technology.

Dr. Azad Madni. Dr. Madni is the CEO and Chief Scientist of Intelligent Systems Technology, Inc. His research interests include enterprise ontologies, enterprise systems architecting, context-aware computing, and modeling and simulation approaches to process design and optimization, integrated product-process development, and distributed training. He has been a principal investigator on sixty-plus R&D projects sponsored by twenty-plus different organizations within DoD, DoC, DoE, MDA, and NASA. He serves on the Industry Advisory Board of the University of Southern California's Systems Architecting and Engineering Program and is also an Affiliate of USC's Center for Software Engineering. He has received numerous awards and honors from both the DoD and commercial sector. He is the only two-time winner of the Developer of the Year Award at the Software Industry Awards in 2000 and 2004. Dr. Madni received his B.S., M.S., and Ph.D. degrees in Engineering from UCLA. He also has an Executive MBA from the AEA/Stanford Executive Institute. He is an elected Fellow of IEEE, INCOSE, SDPS, and an Associate Fellow of AIAA. Dr. Madni is currently serving as the President for the Society of Design and Process Science. He has been a Visiting Industrial Fellow at Caltech's Jet Propulsion Laboratory in the Center for Microelectronics. Dr. Madni is listed in the Marquis' Who's Who in America, Who's Who in Science and Engineering, and Who's Who in Industry and Finance.