Capability Requirements Portfolio Management in Large Organizations using Semantic Data Lake as a Decision Support System: Proof-of-Concept Experiments

Amlan Das

Working Paper CISL# 2018-02

February 2018

Cybersecurity Interdisciplinary Systems Laboratory (CISL)
Sloan School of Management, Room E62-422
Massachusetts Institute of Technology
Cambridge, MA 02142

Capability Requirements Portfolio Management in Large Organizations using Semantic Data Lake as a Decision Support System: Proof-of-Concept Experiments

by

Amlan Das

B.Tech. Information Technology
Visva Bharati University, 2007

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

February 2018

Signature of Author ______________________________
System Design and Management Program
January 18, 2018

Certified by ______________________________
Stuart E. Madnick
John Norris Maguire Professor of Information Technology, MIT Sloan School of Management &
Professor of Engineering Systems, MIT School of Engineering
Thesis Supervisor

Certified by ______________________________
Allen Moulton
Research Scientist, MIT Sociotechnical Systems Research Center
Thesis Supervisor

Accepted by ______________________________
Joan Rubin
Executive Director, System Design and Management Program

(THIS PAGE INTENTIALLY LEFT BLANK)

Capability Requirements Portfolio Management in Large Organizations using Semantic Data Lake as a Decision Support System: Proof-of-Concept Experiments

By

Amlan Das

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering and Management at the Massachusetts Institute of Technology

ABSTRACT

The United States Department of Defense (DoD) is a large and complex organization, which employs a capability based requirements planning process. Decisions on capability requirements are made by senior military officers supported by experienced military and civilian staff with subject matter expertise. There are also many other stakeholders involved in defining concepts, identifying missing capabilities (gaps), evaluating proposed capabilities, recommending solutions to fill gaps, and developing and deploying new and improved capabilities. The process is document-driven. As each document arrives, it is reviewed and a validation decision made. The documents are then filed away. One of the problems faced by the DoD is that, while the documents are retained, the knowledge in the documents is difficult to access except by finding, reading, and analyzing the document again. Abstracting the essential information from documents and storing it as data would enable the staff to make connections from new documents filed to older documents that have related information. Understanding the interdependencies among capability requirements would enable highly informed decisions that are more cohesive with the enterprise strategy for portfolio of systems and capabilities. While there have been incremental steps by the DoD to the decision making process with document repositories and document annotations, there are ways to further improve the process to achieve a full data-enabled, capability requirements portfolio management ability.

This thesis analyzes capability requirements portfolio management challenges, and presents the findings of proof of concept experiments implementing a data driven Semantic Data Lake solution to augment decision support. The data model developed in this research is a hierarchical, linked data model, derived from the specifications for document based information sources, to demonstrate the potential use cases. A semantic data model ontology was built in the Data Lake platform with a selection of realistic data, to validate that it can support the United States DoD architectures and handle the complexity of information interdependency. Semantic Data Lake accounts for discrete data and their relationships, in addition to qualitative influences to facilitate knowledge and fact representation natively. The research findings suggest that Semantic Data Lake can provide the enablers that present the United States DoD architectural information for decision making in a coherent and dynamic way, conducive to draw conclusions that can affect the outcome of the governing of capability requirements.

Thesis Supervisor 1: Stuart E. Madnick
Title: John Norris Maguire Professor of Information Technology, MIT Sloan School of Management & Professor of Engineering Systems, MIT School of Engineering

Thesis Supervisor 2: Allen Moulton
Title: Research Scientist at MIT Sociotechnical Systems Research Center

ACKNOWLEDGEMENTS

I would like to thank all the people who contributed to the work described in this thesis. First, I thank my thesis advisor, Professor Stuart Madnick, for accepting me into his group. During my tenure, he contributed to a rewarding graduate school experience by giving me intellectual freedom in my work, supporting my attendance at various conferences, engaging me in new ideas, and demanding a high quality of work in all my endeavors.

I am particularly thankful to Allen Moulton. He held my hand and guided me all throughout my tenure at MIT, with his vast knowledge and experiences, helping me tackle the murkiness of the "wicked problem" that I was looking into under the systems thinking approach. I greatly benefited from his keen scientific insight, his knack for solving seemingly intractable practical difficulties, and his ability to put complex ideas into simple terms.

I am grateful to all the outstanding Professors, Teachers, and TAs of MIT SDM and Sloan for having the opportunity to learn from the leaders of the knowledge factory of the world. In particular, Professor Bryan Moser, Professor Olivier De-Weck, Professor Bruce Cameron, and Professor Edward Crawley. In addition, the SDM staff has provided outstanding support to me.

Finally, I am extremely thankful to my family for their love and understanding in allowing me to spend the necessary time in my studies.

The work presented here was supported, in part, by the MIT Lincoln Laboratory and the US Army under the "Study of JCIDS Semantic Architecture Framework" project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not reflect the official policy or position of MIT Lincoln Laboratory, the US Army, or the Department of Defense.

Table of Contents

List of Acronyms

AIM	Air Intercept Missile
AMETL	Agency Mission Essential Task Lists
AoA	Analysis of Alternatives
API	Application Program Interface
BCA	Budget Control Act
CBA	Capabilities Based Assessment
CD	Capability Document
CDD	Capability Development Document
CDR	Critical Design Review
CDS	Clinical Decision Support
CDSS	Clinical Decision Support System
CDTM	Capability Development Tracking and Management Tool
CJCS	Chairman of the Joint Chiefs of Staff
CML	Capability Mission Lattice
COI	Critical Operational Issues
CONOPS	Concept of Operations
CPD	Capability Production Document
DAS	Defense Acquisition System
DCR	DOTmLPF-P Change Recommendation
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
DOTmLPF-P	Doctrine, Organization, Training, materiel, Leadership, Personnel, Facilities, Policy
DPG	Defense Planning Guidance
DSKB	Decision Support Knowledge Base
DSS	Decision Support System
EMD	Engineering, Manufacturing & Development
ESAT	Enterprise Strategic Analysis for Transformation
ETL	Extract, Transform and Load
FAA	Functional Area Analysis
FCB	Functional Capabilities Board
FNA	Functional Needs Analysis
FOC	Full Operational Capability
FSA	Functional Solutions Analysis
GBU	Guided Bomb Unit
LCV	Land Combat Vehicle
GEF	Guidance for the Employment of the Force

GUI	Graphical User Interface
HD/CS	Homeland Defense and Civil Support
ICD	Initial Capability Document
IOC	Initial Operational Capability
IT	Information Technology
JCA	Joint Capability Area
JCEA	Joint Capability Enterprise Architecture
JCIDS	Joint Capabilities Integration and Development System
JDAM	Joint Direct Attack Munition
JDSAISR	Joint Direct Support of Aerial Intelligence Surveillance and Reconnaissance
JFC	Joint Functional Concept
JFTL	Joint Future Theater Lift
JIC	Joint Integration Concept
JMETL	Joint Mission Essential Task Lists
JOC	Joint Operational Concept
JROC	Joint Requirement Oversight Council
JSON-LD	JavaScript Object Notation Linked Data
KMDS	Knowledge Management Decision Support
KPP	Key Performance Parameters
KSA	Key System Attributes
LCC	Life-Cycle Cost
MCO	Major Combat Operations
MIT	Massachusetts Institute of Technology
NCBO	National Center for Biomedical Ontology
NDS	National Defense Strategy
NIPRNET	Non-secure Internet Protocol Router Network
NMS	National Military Strategy
NSS	National Security Strategy
OV	Operational View
OWL	Web Ontology Language
OWL DL	OWL Description Logic
PD	Production and Development
PPBES	Planning, Programming, Budgeting and Execution System
QRM	Quadrennial Roles and Missions
R&D	Research & Development
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RDFa	RDF Attributes
RDFS	RDF Schema
RDT&E	Research, Development, Test & Evaluation
SIPRNET	SECRET Internet Protocol Router Network

SME	Subject Matter Expert
SPARQL	Simple Protocol and RDF (Resource Description Framework) Query Language
SRD	System Requirements Document
T&E	Test & Evaluation
TD	Technology Maturation & Risk Reduction
TOC	Total Ownership Costs
UJTL	Universal Joint Task List
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
WIPT	Working-level Integrated Product Team
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

List of Figures

List of Tables

1. INTRODUCTION

"Data is a precious thing and will last longer than the systems themselves." – Sir Timothy John Berners-Lee

The United States achieves its national security missions by equipping its military forces with the best weapons systems in the world. [1] In doing so, all the services and the acquisition system play a critical role in creating capabilities for the defense forces. Creation of capabilities in large and complex organizations like the US Department of Defense (DoD) takes years of careful analysis, strategy formulation, planning and execution. It is the result of timely and exhaustive analysis and documentation of capability gaps and requirements that gives the US military services the strategic, tactical and operational advantage. However, questions arise - what is the envelope of improvement? Can the capability requirements portfolio management and the acquisition system be improved further? Are there synergies, opportunity for collaboration across services and platforms, and critical inter-dependencies, which are not apparent and can severely affect the outcome of the critical decisions being taken as part of the capability requirements portfolio management?

To consider the questions raised above, let's look at how the US DoD manages capabilities, requirements, and acquisitions. The US DoD uses a capability-oriented process to manage activities including requirement process, capability requirement portfolio management, identification of capability requirements and associated capability gaps, development of capability requirement documents, gatekeeping, and staffing procedures. [2] The Joint Capabilities Integration and Development System (JCIDS) was created to manage capability redundancies for efficient capability development to meet the combined needs of all US military services, and to ensure that the capability requirements are validated and verified prior to committing valuable resources on programs. [3] Although documents and data are stored in central repositories, the DoD has not been able to utilize this at an enterprise level - to perform data analysis based on historical data about retired and existing systems and capabilities, or what impact a certain capability has on the entire enterprise architecture, right off the repository. [3] With rapid advances in computing and data technology, it is now possible to consolidate and analyze massive amounts of data. This capability is essential for developing a Joint Capability Enterprise Architecture (JCEA) that can provide an aggregation of multiple silos of information, and bring them together to generate multiple viewpoints for analyses. The JCIDS process has been discussed in details in chapter 2.1, and the JCEA has been illustrated in chapter 2.5.

Additionally, the human resources who handle the acquisition and capability requirements are Subject Matter Experts (SMEs) who rotate in and out of the organization. They bring their in-field knowledge into the organization, scale up to the new organization's processes and procedures, and then move out to a newer organization after a posting of typically one - two years. This means that a new SME replacing the old one has to take diligent hand over, along with transfer of knowledge. The idea behind using a Semantic Data Lake platform is also to encode the "semantics", or knowledge into an easily reusable electronic platform, that can greatly speed up the integration of a new SME and take some of the burden of maintaining the knowledge off to computer systems. A Semantic Data Lake is a repository of raw data saved in the granular native semantic web format, and brought together at query time to present the information. Data Lakes use the concept of pre-processing as little of the data as possible beforehand and literally tosses all the data into the Data Lake in its native form and fishes out what is needed later. Descriptive metadata is where semantics get into the picture. Semantic Technology, covered in section 3, handles the meaning of the data, rather than just the data at the face value. And, semantics added on top of data lakes give – Semantic Data Lake.

A deep dive into capability requirements portfolio management requires data driven analysis of massive amount of existing data to discover insights about underlying effects. Data driven analyses can help take an unbiased consideration into informing better policy and programmatic decisions. [1] Data is the key to making informed and reasoning backed decisions. There are documented evidences of performance improvement with a shift to more data-driven decision-making. [4] Some evidences coming from mathematical modeling of economics also links superior performance with data-driven decision making in a moderate sample of public firms. [4] Data based analysis can help understand why any discernible trends exist. Without identifying trends and forming baselines, it is very difficult to prioritize efforts and initiatives in a rational and optimized manner, and to gauge whether those initiatives are producing the desired results. [1]

One possible lens to look at performance of capability requirements management and acquisition is the amount of Research, Development, Test, and Evaluation (RDT&E) funding spent on major programs that were canceled before producing any units and those that produced many fewer units than originally planned. Figure 1 shows the annual percentage of RDT&E funding (less earlier science and technology expenditures) by each Military Department on major weapon system development efforts with major curtailment or no operational units produced. [1] The Army has the largest chunk of canceled programs. The amount of funding lost was relatively constant for the Army from 2004 through 2010, coming down sharply thereafter. The majority of the Army's sunk funding problem through this period was due to the cancellation of the Future Combat System (FCS). [1] The causes of the program cancellations and curtailments are not examined under this thesis, but it is being used as an example of one of the possible areas of improvements of the overall capability requirements management and acquisition process.

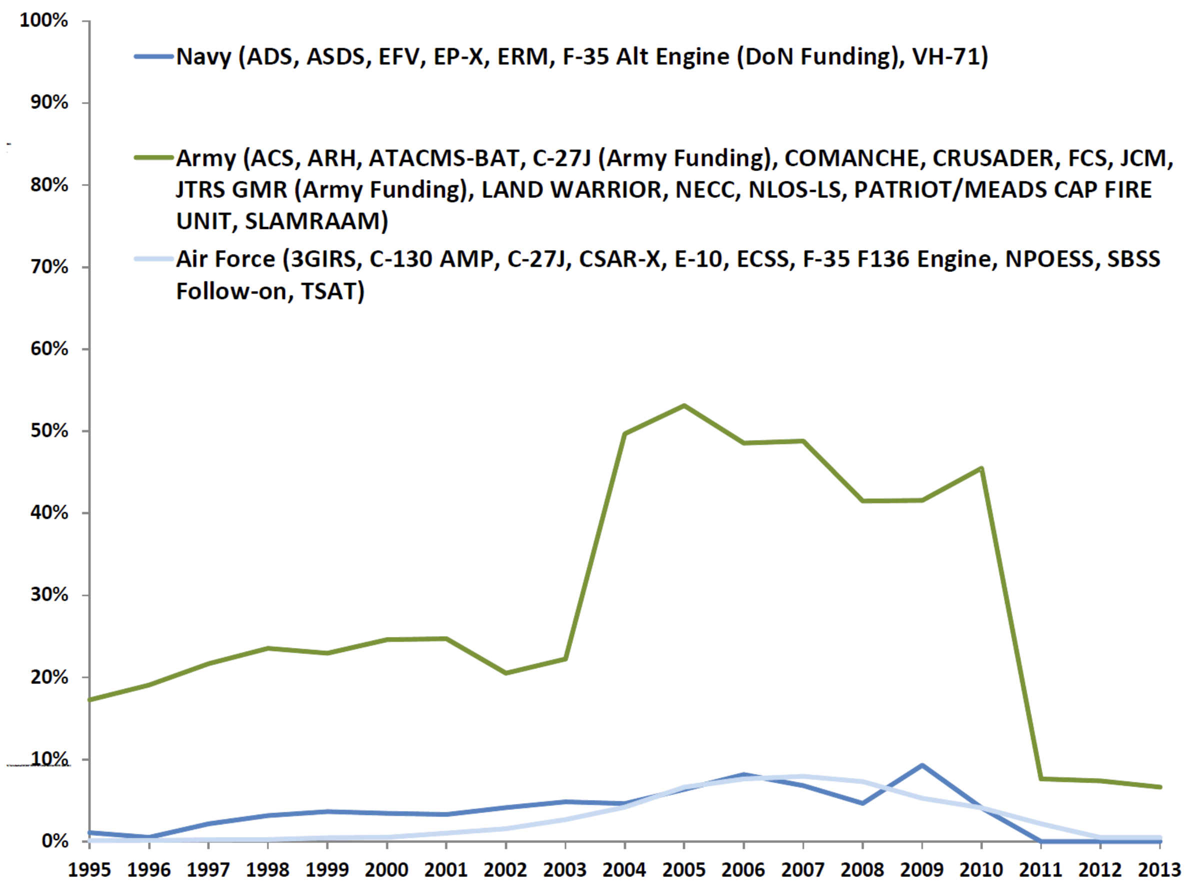


Figure 1 - Major Programs Officially Canceled without Producing Any or Very Few Operational Units (1995–2013) [1]

There are inherent risks involved with developing new and sometimes well beyond the current state of the art weapon systems. These advanced weapon systems provide a decisive operational edge, spanning far into the future. The risks however, can be managed. While expecting perfection is like wishing that the world be ideal, actions for efficient risk management should be able to keep the inevitable problems that will arise within reasonable bounds. [2]

The Joint Common Missile (JCM) and the modified JCM program – the Joint Air-to-Ground Missile (JAGM), have been long running programs (more than 15 years), are analyzed in section 1.1 to discuss some of the evolving problems with capability requirements management, and possible avenue of improvements.

1.1 Analysis of the Joint Common Missile (JCM) / Joint Air-to-Ground Missile (JAGM) Program

The Joint Common Missile (JCM) program, (JCM illustrated in figure 2), was started in 1999 as an air to surface weapon for the allied services' rotary, fixed wing, and Unmanned Combat Aerial Vehicle (UCAV) to neutralize high value fixed, stationary, and moving land as well as naval targets. [5] The JCM started as a joint program between the Army, the Navy and the United States Marine Corps (USMC), and a cooperative development program with the United Kingdom. [5] The JCM was being conceived as an eventual replacement to the HELLFIRE (Air-to-Surface Missile), Tube Launched Anti-Tank Missile (TOW) and Maverick (Air-to-Ground Tactical Missile) families of missiles. [5] The JCM was intended to be a common multi-mode configurable weapon, to be used by the joint forces across various platforms, with a wide range of engagement capabilities, including precision strike and fire-and-forget technologies. [5]

JCM being planned to replace three families of missiles, it was intended to be the weapon of choice for Army rotary-wing systems including the Longbow Apache (AH-64D) and the planned Comanche (RAH-66), the latter of which was also canceled. [5] The JCM was also a candidate for the Future Combat Systems (FCS) ground platforms. [5] The Navy and the USMC was also looking to integrate the JCM with the Super Hornet (F/A-18E/F), the Seahawk (MG-60R), and Super Cobra (AH-1Z). [5]

Figure 2 - Joint Common Missile [5]

On path to be designed as a single missile for all rotary and fixed wing aircrafts, the JCM was to include a package of three different sensors, countermeasures, multi-effects warhead, and a propulsion package that exceeded the capability of existing fielded systems. [5] One of the requirement constraints was for the JCM to be contained within the same physical dimensions as the current generation of HELLFIRE missiles in 2000s. [5] There were several integrative factors at play. The projected JCM load on the planned RAH-66 Comanche would give back 480 pounds to the airframe for increased performance of the aircraft as well as to carry additional fuel. [5] The JCM was in line with the Army Chief of Staff's initiative to achieve first-round kills with small caliber solutions. [5] By providing a common-caliber system for both air and ground use, the JCM aligned with the battlefield commander's operational and logistics flexibility while achieving lifecycle cost savings. [5] The updated threat conditions and aging stockpiles of existing inventory of missiles called for more capable missile systems in the near future. [5] Therefore, the JCM had to be not only had to have incremental capabilities, but also would need to function well with existing legacy platforms.

Lockheed Martin beat Boeing and Raytheon in May 05, 2004 to win an order to develop and build the JCM for the Army, Navy and Marine Corps. [5] However, on December 23, 2004, the Pentagon announced plans for saving of $30 Billion between 2005 and 2011, and among the programs cancelled due to these cuts was the $2.4 Billion funding for the JCM program. [6]

The JCM program was resurrected as the JAGM program in 2008, much with the capability requirements remaining the same as JCM. [7] The JAGM was an Army led program with joint requirements from the Navy and the Marine Corps. The JAGM program continued on the path of technology development phase, and never became mainstream to enter production. The problems faced by JCM-JAGM programs were not directly correlated to cost or to the performance of the program, but rather the JCM-JAGM programs kept being cut to pay for other things. [7] The HELLFIREs was seen as good enough to equip American helicopters and large UAVs like the Predator, mainly to keep the inventory stocked, since the JAGM wasn't production ready yet, in view of the changing world dynamics with the Afghanistan and Iraq wars. There was need for continued stocking of the inventory to tackle the evolving threat conditions, and the JAGM was not ready to go to production. Effective alternatives like the HELLFIREs, smaller guided 70 mm rockets, and the Small Diameter Bomb (SDB-1) glide bombs were being used to fill the operational gaps, and the JCM-JAGM program in its original package of requirements wasn't interesting enough to be continued to be pursued. [7]

After a number of cancellations and reinstatement of the JCM-JAGM program, the US Army looked at the HELLFIRE stocks, and decided to update the JAGM program into an incremental approach, that focused on replacing the most at-risk AGM-114L (HELLFIRE variant) radar-guided missiles first. [7]

The JCM-JAGM program was supposed to equip the warfighters with a capability. When the JCM-JAGM programs were cancelled repeatedly, it left capability gaps unfulfilled. At a high level, it is known that a set of capability gaps were left open. But, what could be the lower level implications of cancelling such a program? The analysis of alternatives can be important in such cases. What would be the alternative solutions for the capability gap? Studying the public sources of data, we can tell that the HELLFIRE family of missiles, or the SDB (Small Diameter Bomb version), were being upgraded in its capabilities, which could then cover some of the capability gaps left by the cancelled JCM-JAGM program. Also, instead of managing the JCM-JAGM program as a single program that would deliver all of the capabilities in one go, what could be the alternative ways of managing the program, e.g., with an iterative, Agile way of delivering incremental functionalities? However, there can be challenges to the incremental Agile way of doing things - the integration of the missile would have to be tested with each different delivery platform (aircrafts, helicopters, UAVs), every time any change is made to the last tested configuration. These are questions, that can be analyzed using a Data Lake platform, and can aid in informed decision making, alternative solution analysis, and to find intersystem dependencies.

1.1.1 Latest JAGM Program Development Plan

The JAGM program was revived in 2012 when the Army restructured the JAGM program to reduce cost and risk. The Army implemented an incremental strategy to deliver JAGM warfighting capabilities over at least three increments. [8] The increments are explained below – [8]

- **Increment one:** The Army plans to develop a dual-mode (laser and radar) guidance section for rotary-wing aircraft. The lethality of JAGM increment one will match that of existing air-to-ground missiles. The new guidance section will be integrated onto the backend of the Hellfire Romeo missile that has an 8-kilometer range.
- **Increment two:** The Army plans to extend the range beyond 8 kilometers and add improved targeting capability. Increment two plans to provide increased operator survivability and improved targeting accuracy over existing air-to-ground missiles.
- **Increment three:** The Army plans to achieve full capability of the JAGM original requirements, improving the missile's accuracy, lethality, and interoperability over existing air-to-ground missiles.

Figure 3 illustrates the JAGM program development increment one. The development plan included a phased incremental introduction of the JAGM system, with the integration of the guidance section into existing Hellfire missiles as part of the initial increment. Appendix A shows the chronological order of key events of the JAGM program from 1999 to 2016.

Figure 3 - Increment One Overview of the JAGM [8]

Figure 4 shows the map of the stakeholders involved in the JAGM program. [8] It is common for multiple stakeholders to be involved in programs of such systems, and thus, stakeholder management in the context of capability requirements becomes imperative. Each stakeholder has a unique perspective and a different incentive to the program. The stakeholders would be better informed about the program if the requisite information they desire out of the program can be presented in a way that is tailored for each party. Information preservation, discovery and presentation can benefit the stakeholders, a capability that can be served by a Semantic Data Lake.

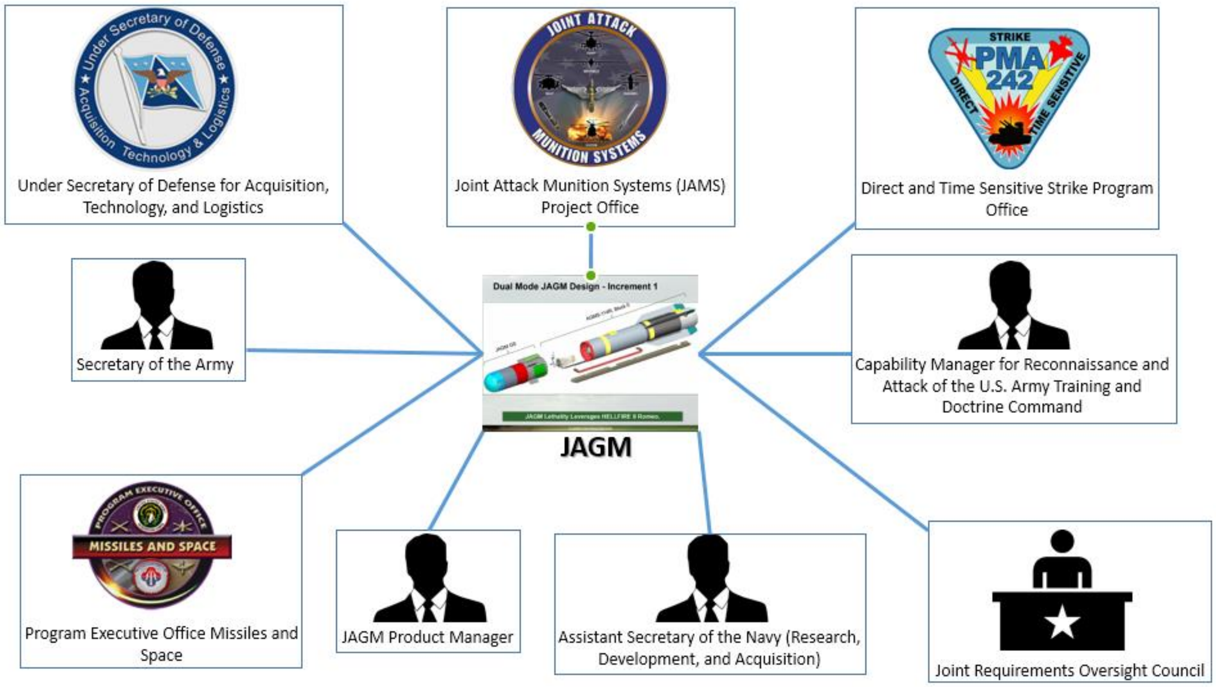


Figure 4 - The stakeholders of the management and oversight responsibilities for the JAGM program

Image Credits - [9] [10] [7] [11]

In 2011, Army and Navy officials concluded that the JAGM program was unaffordable as originally designed because funding was not available to meet program requirements. In 2012, Army and Navy officials determined that they needed to restructure the JAGM program to stay within their available funding. Army officials removed all funds ($136 million) from the FY 2013 JAGM program and decided to develop the JAGM in increments in response to the budget constraints. [8] Navy officials stated that they also removed their funds ($106 million) from the F Y 2013 JAGM program budget. [8]

Therefore, Joint Attack Munition Systems (JAMS) project office officials restructured the program. Specifically, JAMS project office officials lowered the performance of two primary requirements, substituted proven technology for technology still being developed, and deferred the delivery of certain capabilities to future increments to reduce program costs. [8] The Army and the Navy deferred delivery of critical capabilities to future unfunded increments. The Army has not determined the developmental testing and procurement costs for increments two and three. [8]

According to a JAMS project official, stakeholders would re-evaluate the threat situation for the initial production decision and determine whether to continue into production with increment one and begin concurrent development of a future increment. [8]

1.1.2 Key Observations from the JAGM Program

The JAGM program (formerly the Joint Common Missile) started development in 1999. [8] Defense systems are very complex, and typical system development times are between seven and twenty years. [12] Various dynamics of the system context can change during this time. The threat scenario that emanated the capability requirements can change, use context may change, budget constraints can pop up, or effective alternative solutions may be devised.

The JCM-JAGM program runs over a 15+ year period including several cancelations, changes to capability expectations, changes to the alternative systems that might provide the capability, and changes to anticipated operational use and environment. In parallel with those changes, there were swings in funding availability - from competitive needs for the Iraq/Afghanistan wars and from major top line budget cuts after 2011. This left JCM-JAGM competing against other systems (HELLFIRE upgrades, Air Force aircraft with SDB2 glide bombs, for example) that looked to be lower cost, faster to deliver, lower risk at the time. The advantages of a common missile looked to decision-makers to be of lower value vs. those other perceived disadvantages. The extended development time from large requirements, expectations (e.g., unproven tri-mode seeker, need for new warhead design, and extended range) added to the risk of going with JCM-JAGM at each point in time. Funds were repeatedly shifted to upgrades to the HELLFIRE (or to assumptions that no new missiles were needed at the time due to large stocks). As of now, those upgrades to the HELLFIRE will be the basis for the new JAGM (with a more technically feasible dual-mode seeker and the newest HELFIRE warhead design, and rocket motor) that will essentially be the next upgrade to the HELLFIRE family tree. It can be understood from these facts that effective, cheaper, or quick availability of alternate solutions can render a system under development redundant, forcing a change in the developing system's requirement characteristics.

The original JCM was intended to be a modular configurable missile system that was to replace three families of missile systems - HELLFIRE, TOW, and MAVERICK, each with varying capabilities and performance characteristics. For legacy support, the program had to comply with some of the existing HELLFIRE characteristics. The JCM was also planned to be integrated with futuristic Comanche and other existing air platforms - F18, Apache, Super cobra etc. Thus, it was not a program in isolation, but an integration program, with massive capability interdependencies. The JAGM isn't just a missile program, but can be understood as a component in a system of systems environment, which meant it was an integration program too. The missile characteristics, dimensions, and other key features need to be compatible with the platforms it would be mounted on – rotary wing aircrafts, fixed wing aircrafts, and UAVs. All the systems need to come together to deliver the capability desired. The resultant is that one ends up with a whole bunch of capability solution requirements. Addition of every requirement adds cost and time into the program. In addition, analysis of alternatives (AoA) needs to be performed, and the tradeoff models have to be built. The intermingling of various capabilities and their interdependencies makes managing the JAGM program vital in order to be a successful entry into the portfolio of other fielded, legacy, and futuristic capabilities. Capturing information about the program is essential for similar long running programs because the personnel change, requirements evolve, etc. The time varying map of information produced by evolution of the program if recorded and analyzed appropriately can greatly aid in quick decision making for an analyst who is trying to find synergies, find systems compatibility, and other cascading effects across systems. E.g. evaluating the amount of capabilities overlaps and capability gaps from the JAGM and the SDB-2 programs can help the key decision makers with program direction and overall strategy.

1.2 Capability Requirements Portfolio Management

The capability requirements are the set of “needs” identified for the various defense services (Army, Air Force, Navy, Marine Corps, Coast Guard, and National Guard). Every defense service is large, and the capability requirements for each of them are very diverse. The large spectrum of capability requirements brings possibilities of overlaps, and interdependencies among the capabilities. For example, a land based armored combat vehicle can be a common requirement for both the Army and the Marine Corps, with subtle variations. As discussed in the JAGM case above, the missile's capabilities have dependency on the platforms it will be delivered from, e.g. dependency of the weight of the missile to the payload of the aircraft, etc. Similarly, various nuances of the capability requirements such as the operational attributes, the threat context can have direct implications on the performance in critical missions. These portfolios of capability requirements need to be managed effectively in order to find commonality in systems usage, handling the interdependencies, comparing to historical systems and use contexts, as well as finding lineage of systems that have been refurbished or evolved in time with changing requirements and dynamics.

The basic premise here is that if the decision-maker has better, targeted information about a certain set of capabilities, use context, competing alternatives, etc. they can make better decisions – program investments that are less likely to be cancelled. E.g., in the JCM-JAGM program, the program had spent a lot of money and made technology progress, but then the program was cancelled. It was cancelled and reinstated several times. Finally, the program was restarted with a reduced set of requirements for the first increment. And, in parallel, HELLFIREs went through upgrades until the capability of HELLFIREs were somewhat closer to the current JAGM program increment one. Each time the JCM-JAGM program was started, there were millions of dollars spent, but without concrete outcomes. There is a possibility that if the program could have been started as an incremental development program as in its current format, the sunk costs may have been reduced. These are critical decision parameters, and the goal of the Semantic Data Lake is to provide better platform for informed decision-making.

1.3 Integrated view of the DoD Decision Support Systems

In order to discuss the capability requirements portfolio management issues in depth, let's look at the current DoD Decision Support System (DSS).

The Department of Defense has three principal decision-making support systems for meeting its policies and strategy, all of which have been significantly revised over the past few years. All three of these systems are driven by broader policies and strategies created and evolved to meet the missions given to the Department by the country. These systems are the following: [1]

- **Planning, Programming, Budgeting and Execution System (PPBES):** The PPBES is the process to construct plans and programs for strategic planning, program development, and resource determination, in line with the National Security Strategy within the constraints of approved resources.

- **Joint Capabilities Integration and Development System (JCIDS):** The JCIDS process has been established in 2003, and the most recent revision had been released in 2015. This process is used to systematically identify, assess, validate and prioritize gaps in joint warfighting capabilities and determine joint capability requirements to address the gaps. Under the guidance of this process, the US military services generate three main documents - Initial Capabilities Document (ICD), Capability Development Document (CDD), and the Capability Production Document (CPD) to support different phases of development of technology for a set of requirements, and provide the traceability from the needs to the fielded systems. This thesis explores focused solution to augment decision making under the JCIDS process.

- **Defense Acquisition System:** This process is focused exclusively on the weapon systems acquisition, automated information systems and services. Despite being framed on centralized policies, this process has the room to delegate decentralized and streamlined execution of acquisition activities. This approach provides flexibility and encourages innovation, while maintaining strict emphasis on discipline and accountability.

  The intermingling process of the PPBE, JCIDS, and DAS is shown below in Figure 5.

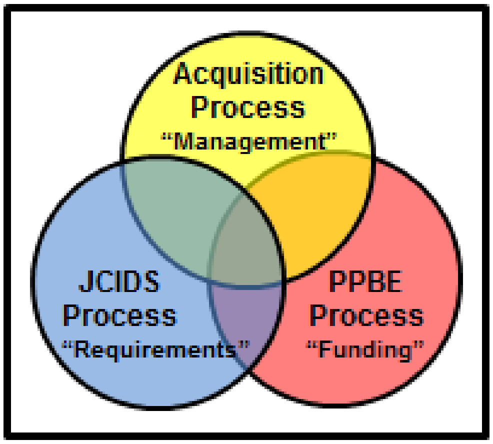


Figure 5 - PPBE, JCIDS, and DAS [2]

The joint military gaps identified for the combatant commanders to effectively execute the strategic, national, operational and tactical missions under the JCIDS process gives rise to the capability requirements. The capability requirements are the needs of the warfighters that, when fulfilled, will plug the gaps identified.

Augmenting the JCIDS process with the capability of data analyses will be the area of focus within the thesis. We will discuss the history and details of the JCIDS process in Chapter 2.

1.4 Scope of Effort

This thesis fits into the larger research effort at MIT to explore and report findings of a suitable data infrastructure platform to manage the portfolio of capability requirements. The direction of the thesis research effort was to explore the use of a semantic data lake framework as a decision support system for the capability requirements management. The thesis seeks to investigate the possible insights that can be gained by building a semantic data lake infrastructure from a common ontology for the use in the analysis of JCIDS documents to support analytics needed for the complexity inherent in the management of joint capabilities.

1.5 Thesis Organization

The remainder of the thesis is organized as follows:

- Chapter 2 (Enterprise Frameworks and Systems in use today with JCIDS) discusses some of the solutions that have been tried, and some of the efforts currently underway apart from the work covered as part of this thesis (Chapter 4) to tame this problem of capability requirements management.

- Chapter 3 (Semantic Technology Overview) explains at a high-level overview, the fundamentals of and core concepts of semantic technology, and what are the building components.

- Chapter 4 (Semantic Technology Decision Support System – Proof of Concept Experiments) explains the effort on the ongoing area of research on building an ontology for the DoD capability requirements management system. It also presents the proof of concept experimental setups, the data that has been used, and the insights obtained from the experimental setups.

- Chapter 5 (Conclusion & Recommendations) presents the learnings and limitations of the current experiments, a discussion on how this methodology can be scaled up to support the DoD, and suggestions for the path to further research.

2. ENTERPRISE FRAMEWORKS AND SYSTEMS IN USE TODAY WITH JCIDS

The JCIDS process was an attempt to manage the capability requirements at an enterprise level. Logical groupings such as Joint Capability Areas (JCA), Universal Joint Tasks (UJT), etc., had been formed for a top-down approach to the portfolio of capabilities and requirements of all the services of the US military. The problem of capability requirements portfolio management is not new, and there have been historical and on-going attempts for an effective solution to this problem. While every attempt is an incremental step towards better managing the capability requirements management, the current theme of increment is the data layer – data management and data analysis. This chapter concentrates on discussing some of the enterprise frameworks and solutions that have been tried, for continual improvement of capability requirements portfolio management.

The MIT Systems Engineering Advancement Research Initiative identifies Enterprise Perspective as one of four key aspects when dealing with engineering systems. [13] The importance of an enterprise perspective cannot be overstated. “In order to successfully design and develop large-scale complex engineering systems, engineers must take all of the enterprise issues into account”. [14] An enterprise perspective will integrate the decentralized diverse information silos, into multiple consolidated views that will help the key people take informed, and data driven decisions.

The goal of the Enterprise Systems Perspective is to frame the enterprise level issues facing the defense requirements process, and help analyze the options available at hand, driven by the strategic direction of an authority, and supported by a decision support system.

The JCIDS was a major attempt to address the shortfalls in the earlier US DoD requirements generation system. The drive to create JCIDS was born out of a memo in March 2002 from the Secretary of Defense to the Vice Chairman of the Joint Chiefs of Staff requesting a study on alternative ways to evaluate requirements. [15] The 2003 JCIDS was created as a formal process to address shortcomings in the requirements generation system. The Chairman of the Joint Chiefs of Staff (CJCS) approved the most recent JCIDS Instruction on 23 January 2015 and its accompanying manual was released on 12 February 2015. [15] The shortfalls in the US DoD requirements generation system prior to JCIDS were identified as: [15]

- Not considering new programs in the context of other programs
- Not sufficiently considering combined service requirements
- Effectively prioritizing joint service requirements
- Not accomplishing sufficient analysis

The central focus of JCIDS is to address capability shortfalls, or gaps as defined by combatant commanders. Thus, JCIDS is said to provide a capabilities-based approach to requirements generation. The previous requirements generation system focused on addressing future threat scenarios. [15] While understanding the risks associated with future threat postures is necessary to develop effective weapons systems, a sufficient methodology requires a joint perspective which can both prioritize the risk associated with future threats and consider operational gaps in the context of all the services. If requirements are developed in this joint context, there is simultaneously a smaller chance of developing superfluously overlapping systems and a greater probability that weapons systems would be operational with one another (i.e. common communication systems, weapons interfaces, etc.). The Joint Capability Areas (JCA) was established in conjunction with JCIDS in order to provide for a common lexicon throughout the US Department of Defense. Another major emphasis of JCIDS is to consider whether a solution to a potential operational gap requires the development of a physical system (a materiel solution) or a procedural or training based solution (a non-materiel solution). In this sense, the JCIDS process provides a solution space that considers solutions involving any combination of Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel and Facilities (DOTMLPF). The Joint Staff supports JCIDS by providing recommendations based on quantifiable data. [15] The Joint Staff collects and analyzes data, and provides observations, findings, conclusions, and recommendations to identify policy, Joint doctrine, Tactics, Techniques, and Procedures (TTP), and materiel solutions and products that promote capability improvement. [15]

The strategic objectives for the functioning of the JCIDS process are laid out in table 1. [16] The primary JCIDS objectives are designated in the third column – Primary.

SL No.	Objective	Primary
1	Identify Joint Military Capability Requirements	X
2	Assess Joint Military Capability Requirements	X
3	Validate Joint Military Capability Requirements	X
4	Prioritize Joint Military Capability Requirements	X
5	Balance Joint Equities	
6	Assist with Informed Decisions	
7	Facilitate Doctrine, Organization, Training, Materiel, Leadership, Policy and Education, Personnel, Facilities, and Policy (DOTmLPF-P)	
8	Drive Defense Acquisition System	
9	Inform Planning, Programming, Budgeting, and Execution Processes	

Table 1 - JCIDS Strategic Objectives [16]

2.1 Decision Making Process under JCIDS

The JCIDS process supports the Acquisition Process by identifying and assessing capability needs and their associated performance criteria. A depiction of the relationship between the JCIDS process and key acquisition process decision points (Milestones and Phases) are shown in the figure 6 below. [17]

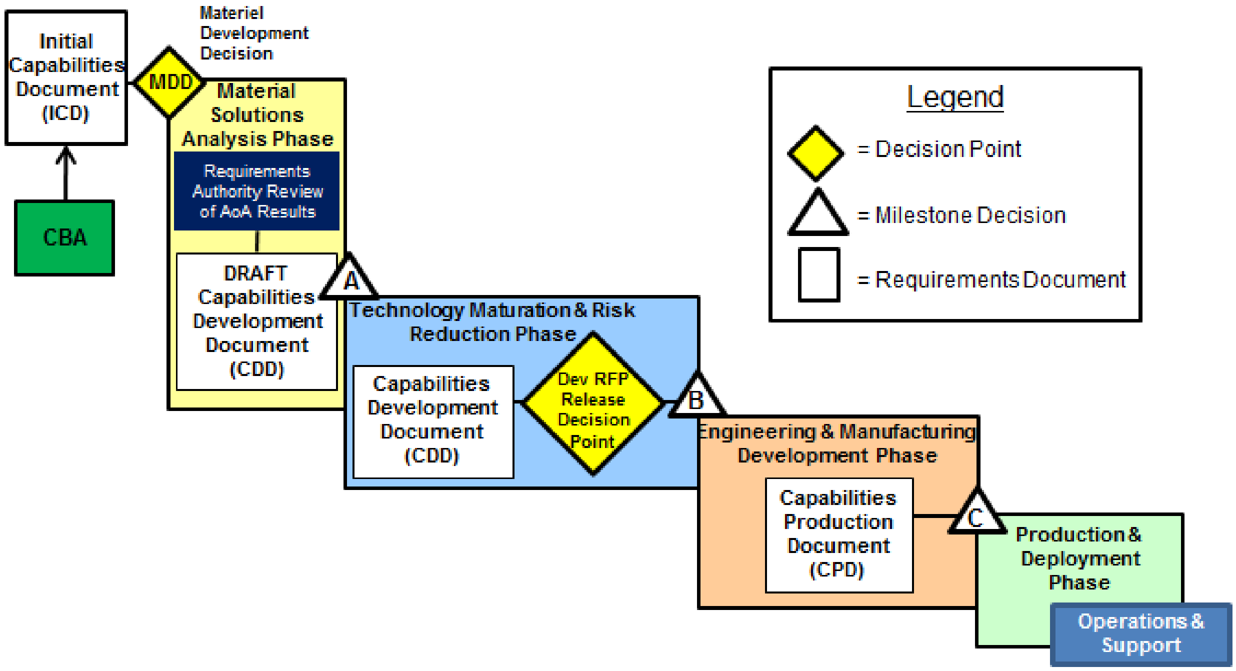


Figure 6 - JCIDS Process within the Acquisition Process Phases & Milestones [17]

The decision making process depicted in figure 6 comprises of 4 major steps – Capabilities base Assessment, Approval of the ICD and Courses of Action, Approval of the Capability Development Document, and Approval of the Capability Production Document. The detailed explanation of these is given in the subsections that follow.

2.1.1 Capabilities Based Assessment (CBA)

The Capabilities Based Assessment (CBA) is the analysis portion of the Joint Capabilities Integration and Development System (JCIDS) process that is performed by sponsoring organizations before submitting documents for higher-level review and validation. The CBA provides recommendations to pursue a materiel or non-materiel solution to an identified capability gap that meets an established capability need. The CBA analysis contains the Functional Area Analysis (FAA), Functional Needs Analysis (FNA) and Functional Solutions Analysis (FSA). This analysis is meant to: [18]

- Define the mission
- Identify capabilities required
- Determine the attributes and standards of the capabilities
- Identify gaps
- Assess operational risk associated with the gaps
- Prioritize the gaps
- Identify and assess potential non-material solutions
- Provide recommendations for addressing the gaps

An approved ICD, written from where the CBA leaves, is the outcome of the JCIDS process. The CBA provides general recommendations to the direction of the type of material solution - whether an information technology system, incremental improvement to an existing capability, or a transformational capability right from scratch. The CBA defines the operational framework from the Combatant Commander's priorities, to evaluate alternate material and sustainment solutions. The CBA also provides affordability advice to the PPBE Process. All CBAs are based on strategic guidance documents. [18]

2.1.2 Approval of the Initial Capabilities Document (ICD) and Courses of Action

The Initial Capabilities Document (ICD) captures the approach to a specific capability gap. The approach can be a material one, a non-material, or a combination of materiel and non-materiel approaches. The ICD defines the capability gap in terms of the functional area, the relevant range of military operations, desired effects and time. [19] The ICD summarizes the results of the Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel, and Facilities (DOTMLPF) analysis and describes why non-materiel changes alone have been judged inadequate in fully providing the capability. [19] The ICD can be used to establish boundary conditions for the scope of alternatives to be considered in the subsequent Analysis of Alternatives (AoA). Concept Refinement and Technology Maturation & Risk Reduction (TD) phase is guided by the ICD. The ICD also supports the Analysis of Alternatives (AoA) and the Milestone A (Figure 6) decision. The ICD is not updated once approved. This is because the analysis does not change, although the approach to arrive to a solution may undergo several iterations and updates. The ICD defines the gaps in terms of the functional area, Range of Military Operations (ROMO), desired effects, time and DOTMLPF, as well as policy implications and constraints. [19] The outcome of an ICD could be one or more DOTMLPF Change Recommendations (DCRs) or Capability Development Documents (CDD). Given below is the format required for a JCIDS Manual Required ICD Format. [19]

- Concept of Operations (CONOPS) Summary
- Joint capability Area
- Required Capability
- Capability Gaps and Overlaps or Redundancies
- Threat and Operational Environment
- Ideas for non-material approaches (DOTMLPF Analysis)
- Final recommendations
- Mandatory Appendices
- Appendix A: Integrated Architecture Products
- Appendix B: References
- Appendix C: Acronym List

2.1.3 Approval of the Capability Development Document (CDD)

The Capability Development Document (CDD) is the phase where the capabilities and requirements take the form of a system. A single capability increment is defined in the CDD as part of the evolutionary acquisition system. The CDD specifies the operational requirements of the specified or conceived system that will fulfill the capability to meet the operational performance criteria specified in the ICD. The CDD has thresholds and objectives used to describe incremental capability. The CDD is prepared during the Technology Maturation & Risk Reduction Phase to guide the Engineering, Manufacturing & Development (EMD) Phase by defining measurable and testable capabilities. The CDD supports and must be validated and approved before the Milestone B (Figure 6) decision at the Development RFP Release Decision

Point. The CDD states the operational performance attributes of the proposed system. Each program increment has a separate CDD or a separate annex from a master CDD. Various tangibles like the Key Performance Parameters (KPP), Life-Cycle Cost (LCC), and Total Ownership Costs (TOC) are introduced in the CDD. The Test & Evaluation strategy is laid out with details like the specific, desired, and operational capabilities, Critical Operational Issues (COI), identifying resource requirements more precisely, etc. [20]

The CDD required sections as per JCIDS manual: [20]

- Operational Context
- Threat Summary
- Capability Discussion
- Program Summary
- Development KPPs, KSAs, and APAs
- Other System Attributes
- Spectrum Requirements
- Intelligence Supportability
- Weapon Safety Assurance
- Technology Readiness Assessment
- DOTMLPF-P considerations
- Program Affordability

2.1.4 Approval of the Capability Production Document (CPD)

The Capability Production Document (CPD) is the preparation step towards full-scale production of the system delivering the capability increment. Information necessary to support production, testing, and deployment of the capability increment is captured in the CPD. The specific attributes that contribute to the most desired operational capability is identified in the CPD. The CPD is prepared during the Engineering & Manufacturing Development (EMD) Phase to guide the Production and Deployment phase after the Critical Design Review (CDR) and is used to measure the contractor's delivery. The CPD is required for the Milestone C Review (Figure 6) and must be certified prior to a program proceeding into the Production and

Development (PD) Phase. The CPD refines the KPP and the performance attributes from the CDD. There will be a separate CPD for a program with multiple increments. [21]

The JCIDS manual required CPD format is shown below: [21]

- Executive summary
- Capability Discussion
- Analysis Summary
- CONOPS Summary
- Threat Summary
- Program Summary
- System Capabilities required for the Current Increment
    - Key Performance Parameters
    - Thresholds
    - Objectives
- FoS and SoS Synchronization
- IT and NSS Supportability
- E3 and Spectrum Supportability
- Technology Manufacturing Readiness Assessment
- Assets Required to Achieve Full Operational Capability (FOC)\
- Schedule and Initial Operational Capability (IOC) and FOC definitions
- Other DOTMLPF and Policy Considerations
- Other System Attributes
- Program Affordability

2.2 Shortfalls of the Current JCIDS Process

To understand the capability requirements portfolio management challenges and issues under a use case such as the DoD, it is critical to understand JCIDS' operating environment. The Joint Requirements Oversight Council (JROC) assists the Chairman of the Joint Chief of Staff (CJCS) in "assessing, prioritizing, and approving joint military requirements". [22] The JROC takes into consideration cost, schedule, and performance trades to ultimately shape the force of today into the force of tomorrow. [16]

Requirements are vetted and approved using JCIDS. Nine strategic objectives were identified for the JCIDS process (Table 1), of which the primary objectives are to ensure the capabilities required by the joint warfighter are identified, assessed, validated, and prioritized in a transparent process that allows for a balanced and informed decision. JCIDS calls for the Prioritization of Military Capability Requirements (Table 1, objective 4). The intent of the JCIDS process is to employ a synchronized, collaborative, and integrated approach that links strategy to capability. [16]

Some of the criticisms of JCIDS: [16]

1. Solution development and delivery are not timely
2. Decisions are made late to need or with poorly scoped information (This thesis attempts to address this point)
3. The process is complex, cumbersome and too document-centric (This thesis attempts to address this point)
4. It lacks mechanisms to focus review across portfolios (This thesis attempts to address this point)
5. It does not control “requirements creep”
6. It does not include key customers in the decision process
7. It does not have tracking mechanisms to trace developments from gap identification through solution fielding (This thesis attempts to address this point)

The Semantic Data Lake platform for decision support system proof of concept experiments, developed in the scope of this thesis seeks to address 2, 3, 4, and 7 from the JCIDS criticisms mentioned above. Viz. – late decision making with poorly scoped information, JCIDS process being complex and document centric, that JCIDS lacks mechanisms to focus review across portfolios, and that JCIDS does not have tracking mechanisms to trace developments from gap identification though solution fielding.

Improvement of the second point – “late decision making with poorly scoped information”, is a straightforward goal of the Semantic Data Lake. The intention of using data broken down into its raw constituents and then using the various information sources to deep dive, compare and contrast capability would alleviate the information barrier with decision-making. Key people should be able to dig out relevant information from a single repository, without many efforts as compared to manual work or other solutions already in place. This will help the analysts focus on the core work of enhanced decision-making backed by data.

The third point – JCIDS process being complex and too document centric is why the Joint Capability Enterprise Architecture (JCEA) and the Semantic Data Lake is being proposed. The JCEA strives to bring together disparate architectures under one umbrella to be able to get a hawk eye view of the whole enterprise. The Semantic Data Lake intends to move away from documents being the primary source of information retrieval for decision-making, to a data centered approach. Documents would still be written as per the JCIDS process, but the Semantic Data Lake is designed to break apart the “semantics” of the document into granular data, concepts, and data models, and bring these together at query time to find the most relevant answer to pressing questions. Moving from a document store to a data-store that has linkages to the source documents can ease the job of sifting through documents, by letting the machine do the job of figuring out the most relevant information connected to a capability, concept, or architecture, and the user (analyst) can look back into the exact documents if required.

The fourth point – “lacks mechanism to review across portfolios”, is partly a technology problem, and partly an organizational one. The technology problem can be addressed by using a Semantic Data Lake. The technology barrier to review across portfolios was the unavailability of a unifying architecture and repository that supports the architecture. Semantic Data Lake with data models that support the JCIDS-JCEA architectures can conjoin diverse systems and capabilities based on common concepts such as Joint Capability Area (JCA), Universal Joint Tasks (UJT), operational attributes, etc. Such capability would give the decision maker the tools at hand to view from an enterprise perspective and evaluate the whole portfolio of capabilities and requirements.

The seventh point – "does not have tracking mechanisms to trace developments from gap identification through solution fielding" is another motivation behind deriving a solution – Semantic Data Lake. This is partly because analysis is still being done with JCIDS documents. Tracking mechanism can become very unsystematic and unreliable if a user needs to go back to documents, sometimes hundreds of pages long to connect each concept back to the gap identification stage. Moving the critical semantics of the documents into a Semantic Data Lake would put the onus of connecting lineage of capabilities and requirements on the data platform. This is where the intrinsic nature of a Semantic Data Lake fits this use case perfectly. Semantic Data Lake has inherent links and connections in the form of Universal Resource Identifiers (URI) to other data objects. Since essential information is captured from document into raw data inside a Semantic Data Lake, linkages between data objects can be created, destroyed, or updated. These linkages can help track capabilities and requirements from "cradle to grave" right from gap identification through fielding of solutions and systems.

2.3 Knowledge Management/Decision Support (KM/DS) System

The Knowledge Management/Decision Support (KM/DS) is a system that acts as a repository for documents that are created via the JCIDS process. The intent of having a KM/DS like system is to have the documents versioned, searchable, and annotated appropriately to provide the ability to tag, search, and retrieve documents pertaining to a certain search criteria. The idea here is to annotate the documents with tags to create metadata about the document, for better document traceability and connect documents to concepts. The KM/DS is still however remains a document centric approach.

"The KM/DS system is the authoritative system for processing, coordinating, tasking, and archiving capability requirement documents, validation memorandums, and related data when classified at or below the level of SECRET." [23]

KM/DS is a document repository with limited metadata and document discovery ability. Efforts are currently underway to add metadata and make document discovery more effective, via new technology and a project to add metadata to the documents in the KM/DS. The KM/DS system still however doesn't capture the "semantics" of the contents of the documents. E.g., The

JAGM program is connected to the Universal Joint Task List (UJT) TA 3, which means employ fires. This UJT – TA 3 is applicable not just for the JAGM system, but there can be various other capabilities or existing systems that use the UJT - TA 3. This type of granular information and their inherent interdependencies cannot be captured under the current KM/DS solution.

2.4 Capability Mission Lattice (CML)

The Capability-Mission Lattice (CML) was introduced as a construct to facilitate the integration of the many factors that must be considered in the identification, assessment, and validation of capability requirements and the associated capability gaps.

Figure 7 shows the CML version 2.0 (latest version) that provides an integrating construct to ensure traceability to strategic guidance, missions of the Joint Force, and other departmental activities - both in the identification of capability requirements and their associated gaps, and in the review and assessment of capability requirement portfolios. The CML incorporates existing JCIDS taxonomies, such as the JCAs, as well as extending into other pertinent areas of the requirements domain. [16]

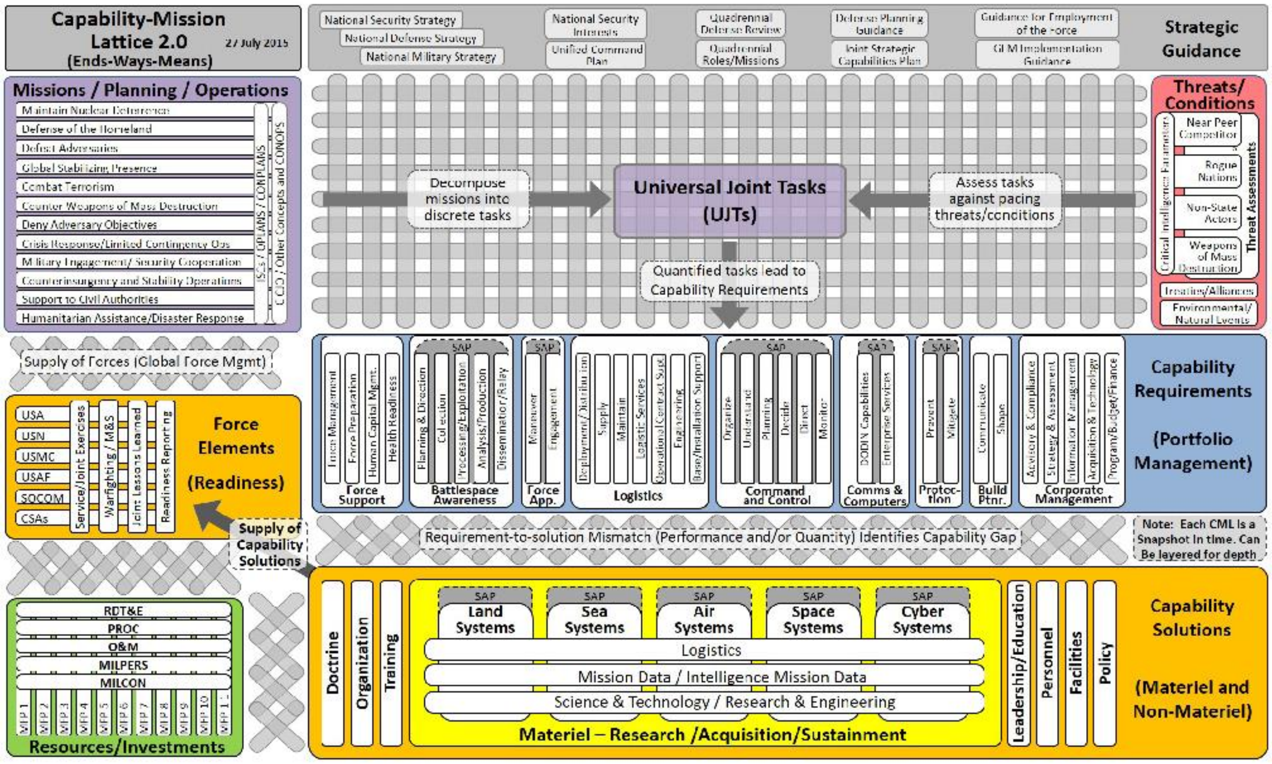


Figure 7 - Capability Mission Lattice (CML) V 2.0 [24]

The CML diagram is composed of ten interlocking sections:

1) Middle left, cross hatched, Supply of Forces (Global Force Management) – This is the purpose of the entire requirements and acquisition process, the supply of necessary forces to Combatant Commanders in the field to support missions assigned by national leadership.
2) Middle left, yellow color, Force Elements (Readiness) – These are the ready to use resources developed and maintained by the Services (Army, Air Force, Navy, Marine Corps) and other Defense components that are available for provision to the combatant commanders to execute missions.
3) Bottom, yellow color, Capability Solutions (Materiel and Non-Materiel) – This is the form the solution will take, e.g., a new airplane, a new software load to an existing system, or training a new way. [16]
4) Lower left, green color, Resources/Investments – These are the budgetary categories that supply funds for developing and maintaining ready capabilities, e.g., RDT&E, Procurement, etc.

5) Top middle, gray color, Strategic Guidance – These national strategy guides the overall capability requirements portfolio, e.g., National security strategy, National Defense Strategy, National Military Strategy, etc.
6) Top left, purple color, Missions/Planning/Operations – These are the high-level mission categories that need to be planned for e.g., Defeat adversaries, combat terrorism, etc.
7) Top Right, red color, Threats/Conditions – These exogenous factors influence the requirements space, e.g., Weapons of mass destructions or natural events. [16]
8) Center, purple and gray color, Universal Joint Tasks (UJT) – These represent a decomposition of mission and operation types (from the left side) into discrete tasks matched against threat factors (from the right side) using the hierarchical task list called the Universal Joint Task List (UJTL), e.g., Deploy and Employ Mounted Forces or Support Vertical Maneuver. Tasks are quantified by the performance level required to achieve the required result given threats.
9) Middle, blue color, Capability Requirements (Portfolio Management) – This is how the forces and capabilities are governed. The capabilities are hierarchically categorized as Joint Capability Areas (JCA), e.g., Force Application or Battlespace awareness.
10) Middle, gray cross hatch, between blue and yellow sections, Requirement-to-Solution Mismatch, - This identifies quantitative or qualitative gaps between capabilities that are required and the portfolio of solutions that are available. Gaps identify where investment may be needed.

The Missions/Planning/Operations section has missions that are decomposed into discrete tasks that are hierarchically arranged under the UJTs or comparable Service Task Lists (as a decomposition at the bottom of the UJTL hierarchy). These Tasks are matched against the prevailing threats and external conditions, and are updated with the changing dynamics of the world. The Tasks match up to the JCAs that provide the frameworks for the capability requirements. This means that every capability requirement has at least one JCA and one Task assigned to it. The capability requirements are fulfilled using capability solutions, which can be material, or non-material, fulfilled by the resources and investments, to supply capability solutions to the services/force elements.

The two cross hatched connections (1 and 10) are critical. Connection 10 ties requirements to solutions and identifies areas where investment should be considered. Connection 1 captures the purpose of the entire system – providing military resources to the field commanders charged to carrying out missions assigned by national leadership (whether fighting campaigns and battles or supporting humanitarian efforts). Every step in the requirements process leads to supporting Connection 1.

2.5 Joint Capability Enterprise Architecture (JCEA) to Augment the JCIDS Process and Address the JCIDS Gaps

The JCIDS process is a warfighter's side process, which is formal, bureaucratic and document driven. The documents are the artifacts of the formal process. The Services and other Defense Components, who are in charge of maintaining adequate capabilities to meet national needs, go through their own processes of determining and documenting what they need. They then write Capability Requirements documents that are then submitted for approval according to the JCIDS process as shown in Figure 6. Once an ICD has been formulated and approved, this is then weighted and analyzed by the Acquisitions team to decide on the course of action to be taken for the ICD, which produces the CDD. The CDD contains information about what solutions should to be acquired.

2.5.1 Existing Capability Based Approach

The capability-based approach is built upon the concept of what capabilities are needed in the end, not what specific systems would provide that capability. The idea of capabilities is to take the focus away from the system desired to the capability desired. For example, instead of discussing that we need to buy a KC-46 midair tanker, the locus of attention should be that there is a need to be able to refuel aircraft midair. That is what the ICD states – the capability desired – and that is the premise of later CDDs and CPDs. To dig deeper, the refueling need arises because there are not going to be enough tankers to do the job in the future. This could be due to various reasons from end-of-life of existing tankers, to the need of technology refresh.

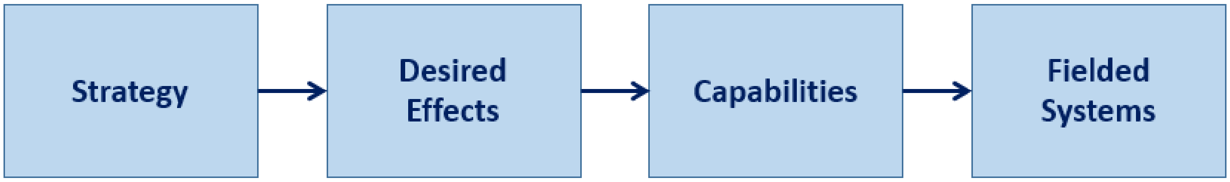


Figure 8 - Value Proposition for Capability-Based Planning (Aldrich Study, 2004) [25]

Figure 8 above shows the value proposition for capability based planning approach. This concept is what JCIDS has been created for. It starts with the national strategy, which directs what desired effects are to be created. The desired effects in turn help deduce what capabilities are required to attain those effects. The required capabilities trickle down to the systems that would empower those capabilities.

Once the ICD is approved, it goes through a detailed analysis to check if it qualifies for a material development. This phase is called the Material Development Decision. A very high-level overview of the process is shown in figure 9 below.

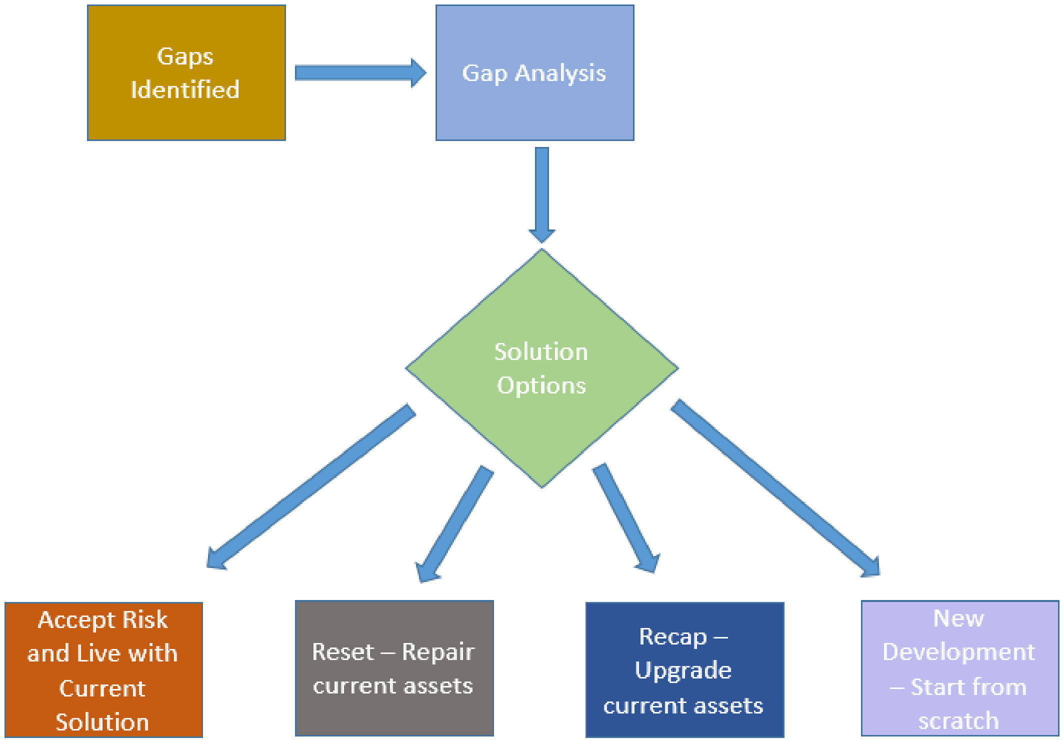


Figure 9 - Solution Options outcome from Gap Analysis

If the solution option path chosen is the new systems development or in some cases even the recap, the desired capability is then validated and an initial contract is issued for technology maturation and risk reduction as milestone B in figure 6. At the end of the process, a design review is conducted and a CDD is written. The CDD is then sent back to the Joint Requirements Oversight Council (JROC). The CDD contains some design outlines with enough design space for engineers, with Key Performance Parameters (KPP) definitions for the intended system. This is then put into a contract in the form of a Request for Proposal (RFP). When the CDD is approved, this capability requirement now becomes a potential program, which is called a package.

2.5.2 Capability Portfolio Management with Semantic Web Tools

Figure 10 below summarizes the main limitations under the current JCIDS process.

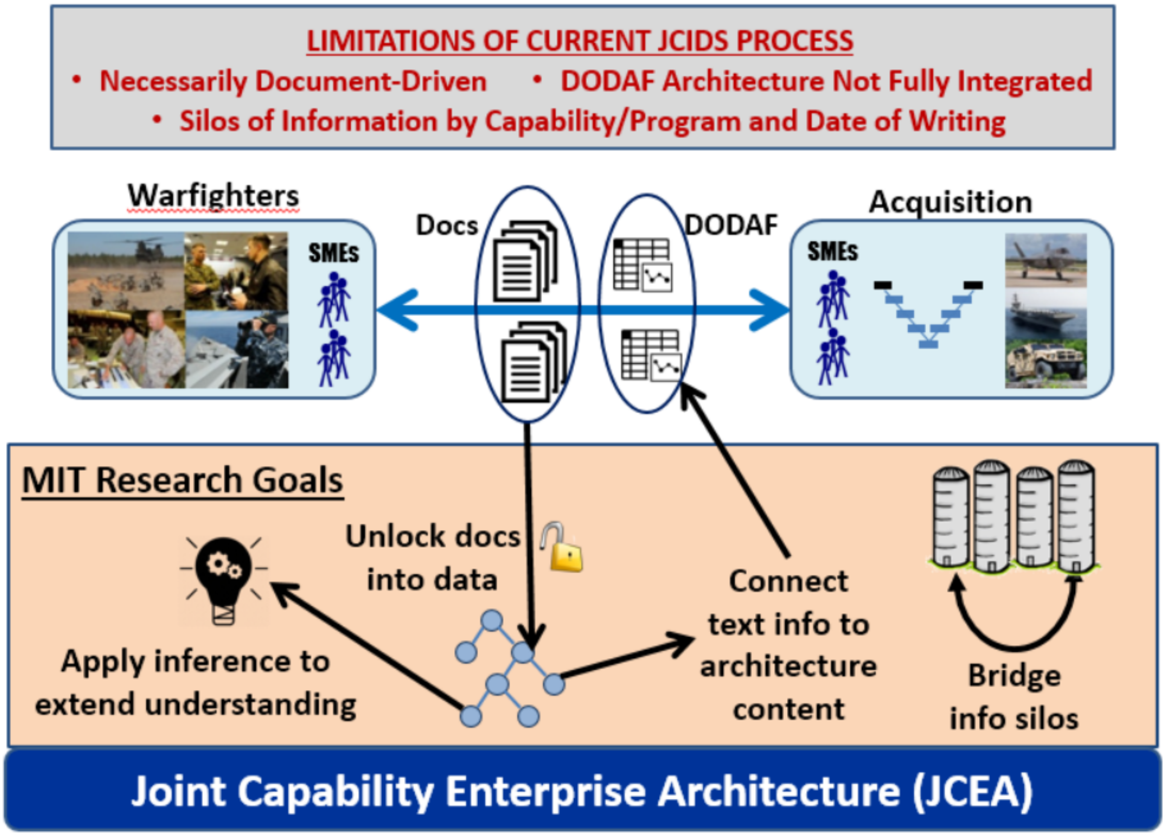


Figure 10 - Joint Capability Enterprise Architecture to support the JCIDS process [25]

The important part of the capability portfolio management as shown in figure 10 is that the warfighters and the acquisition departments have to communicate with each other. Documents and Department of Defense Architecture Framework (DoDAF) are the medium of interaction between the acquisition side and the warfighters. There is a continuous back and forth between the Functional Capabilities Board (FCB), Subject Matter Experts (SMEs) and the service staff (warfighters), facilitated by the documents. All these documents are filed in the KM/DS document repository. But the KM/DS has limitations as discussed in chapter 2.3. Moreover, the KM/DS solution is still essentially document based, not data-driven. If one document deals with a certain capability requirement, it is very difficult to determine systems with similar capabilities in service, or other related capabilities. In addition, the "semantics" of the data, i.e. the meaning of the content of the documents are locked inside the silo of the document. An example would be the Universal Joint Task (UJT) that applies to a certain capability. In order to find UJT interdependencies among different systems (like in the case of JAGM and SDB), one would need to sift through documents to map it out. The ICD Ontology explained in Appendix C, is implemented as part of the PoC experiments, lays out connections between programs and systems based on concepts like UJT. A semantically based solution can free up the data from confines of the document, and make it exposed, to be searchable and analyzable to find connections and interdependencies. The semantic architecture strategy takes the documents apart into chunks of data. The original MIT research goals included applying logic rules engine (inferencing) to extend the understanding of relationships from the raw data in the Semantic Data Lake. The software used during the experiments had limited scope of integrating rules engine. However, this thesis demonstrates the logic rules encoded manually as Semantic SPARQL queries. Proving the capability of using an inferencing engine can be scoped out for possible future research. The prototype ICD requirements ontology framework included in Appendix C provides a semantic structure, which enables data chunks extracted from the documents to be organized by semantic role in the capability requirements analysis. Extracted data, tagged with ontology concepts, captures essential features of document contents from the original DoD documents – ICD, CDD etc., which can then be loaded in a semantic graph database as described in Chapter 4. The semantic graph database will now set the ground to perform portfolio management. The intention is to expose the semantics of the silos of the documents, to help better manage the portfolio of capability requirements.

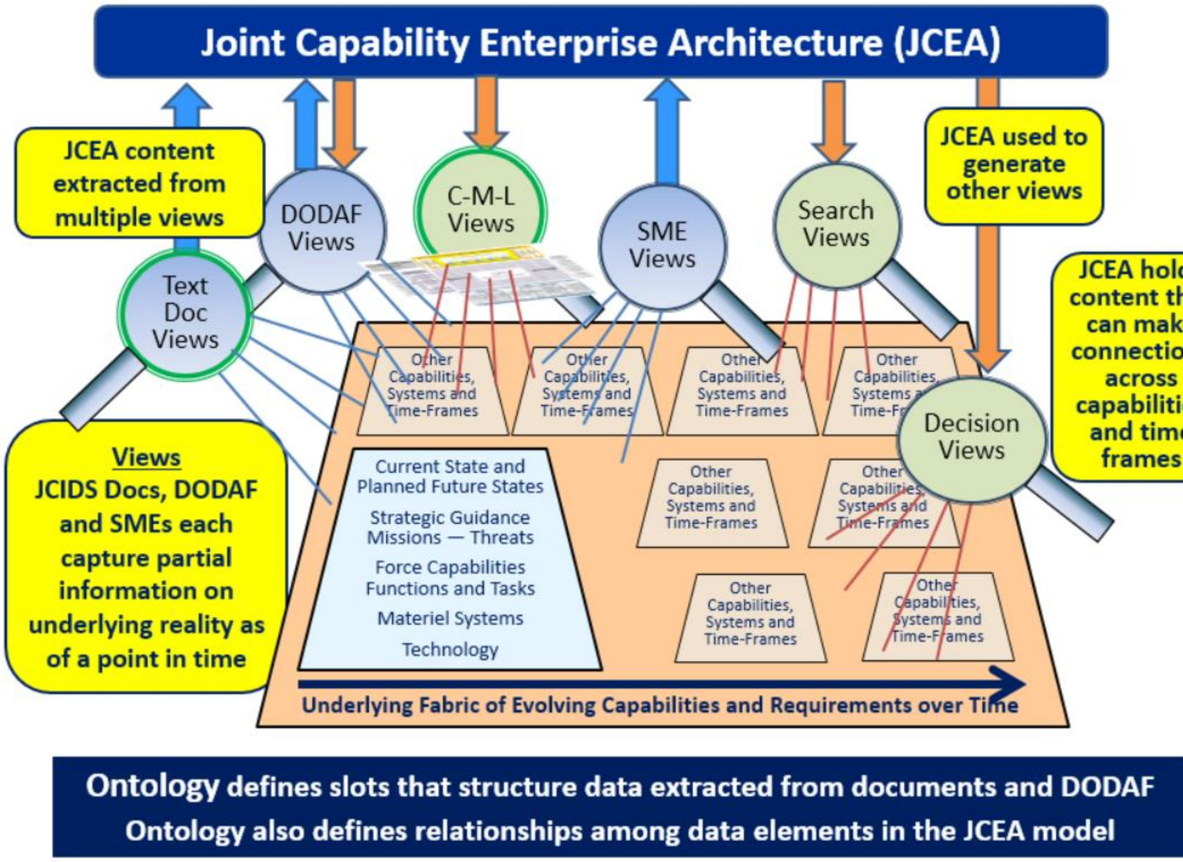


Figure 11 - Joint Capability Enterprise Architecture [25]

There can be multiple stakeholders to a single program or capability as was depicted in figure 4 (chapter 1.1.1) for the JAGM program. Each stakeholder may want to generate a unique view of the ecosystem for the portfolio of capabilities. Figure 11 depicts how different viewpoints can be generated from a single Data Lake that aggregates information from various sources. Figure 11 also shows the concept of the Joint Capability Enterprise Architecture, JCEA. [3] The JCEA aims to inter-relate and inter-connect information fed into the JCEA generated from various formal documents. The JCIDS documents, DoDAF documents, and those generated by the SMEs capture information about related systems, but there is no standard way to figure out these interconnections and inter-relations. The goal is to extract the information contained in various documents, enrich them with other data from SMEs and the various departments, and create a data lake that can serve the purpose of a single repository of information about the system of systems environment in the DoD. JCEA would then be able to generate specialized viewpoints for the intended type of user. A data lake is a storage repository that holds a vast amount of raw data in its native format until it is needed. E.g., The DoDAF view can be separate from the SME view; however, they would be scraped out of the same massive data lake, i.e. the data lake would serve as the raw data repository, and specialized queries would be used to present the information, the way user desires. It would also be possible to relate viewpoints back to the original source or document(s) that they came from. This is an additional advantage of using such a system as it can provide a macro view of how a system or other related systems have evolved over time.

3. SEMANTIC TECHNOLOGY OVERVIEW

Semantic Web is a mesh of information linked up in a way so that it can be easily processed by machines in a large scale. It can be described as a globally linked database. Semantic Technology encodes meanings separately from data and content files, and separately from application code. This enables machines as well as people to understand, share and reason with them at execution time.

With traditional information technology, on the other hand, meanings and relationships must be predefined and "hard wired" into data formats and the application program code at design time. This means that when something changes, previously un-exchanged information needs to be exchanged, or two programs need to interoperate in a new way, the humans must get involved. [26]

The semantic web standards have been created not only as a medium in which people can collaborate by sharing information, but also as a medium on which people can collaborate on models. Models that they can use to organize the information they share; models that they can use to advance the common collection of knowledge. Models are used to organize human thoughts in the form of explanations.

The Semantic web provides a number of modelling languages that differ in their level of expressivity. They constitute different tools that allow different people to express various types of information. The Semantic Web standards are organized so that each language level builds on the one before such the languages themselves are layered.

There are four guidelines for semantic web data, or linked data shown in table 2 – [28]

Number	Guideline
1	All objects and names are expressed in URI
2	URIs should be HTTP URI so that this information can be looked up
3	URIs should provide useful information
4	Include links to other URIs so that more information can be discovered by the user

Table 2 - Four Guidelines for building Semantic Web / Linked Data [28]

Semantic Web technology is based on the concept of Resource Description Framework (RDF), which is a triplet of information. It is illustrated in section 3.1.

To illustrate the model of the semantic web – linked data, an example is shown in figure 12 – two relational tables of two different databases.

movie_id	title	description
.	.	.
.	.	.
.	.	.
47236	Pulp fiction	
53821	Ace Ventura	

theater_id	name	address	movie
.	.	.	.
.	.	.	.
.	.	.	.
11124	Pavilion Cinemas	188 Prospect Park West, Brooklyn, NY, USA	Pulp Fiction
98321	Cobble Hill Cinema	265 Court Street Brooklyn, NY, USA	American Graffiti

Figure 12 - Two relational tables showing movie information and theater hosting the movie [28]

The above information model can be encoded in linked data and following the four guidelines shown in table 2. The data represented in a linked data format is shown in figure 13.

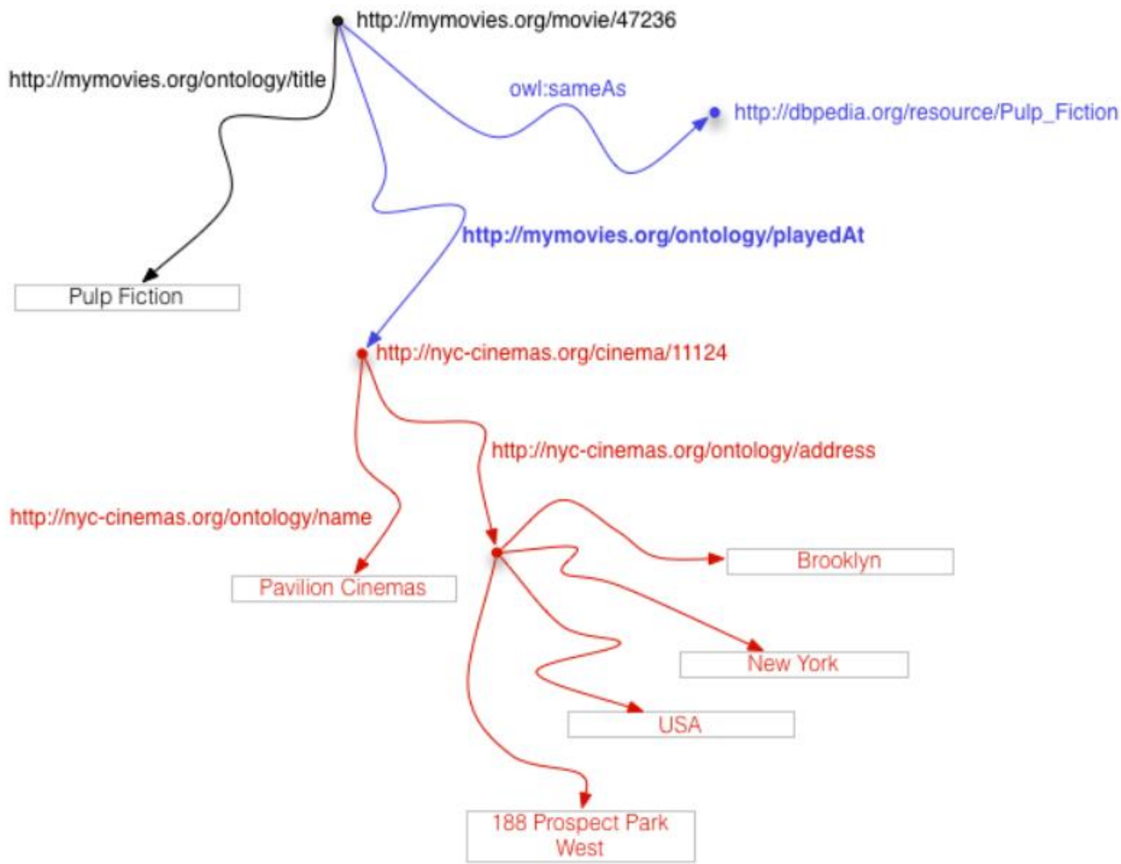


Figure 13 - Graph view of movie database in figure 12 represented as linked data [28]

The various modeling languages under semantic web technologies are described in sections 3.1 to 3.4

3.1 RDF – Resource Description Framework

An RDF datastore, also known as triplestore is a directed labelled graph, where the edges denotes the tagged link between two resources, represented by graph nodes. A triple is a data entity composed of subject-predicate-object structure.

RDF is the basic building block of the Semantic Web. The directed edges in an RDF are labels, or rather explicitly, Universal Resource Identifiers (URI).

There are 3 types of nodes in an RDF directed graph: [29]

a) **Resource Nodes:** A resource is anything that can have things said about it. A resource can be imagined as a thing vs. a value. In a visual representation, the resources are represented by ovals.

b) **Literal Nodes:** A node that contains a value is called a literal node. Example of a literal node can be any name, e.g. John. In a visual representation, literals are represented by rectangles.

c) **Blank Nodes:** A blank node is a resource without a URI. The resource represented by a blank node is also called anonymous resource.

Edges can go from any resource to any other resource, or to any other literal, with the only restriction being that edges cannot go from literal to anything at all. The linking between things is the fundamental capability of the Semantic Web, and is enabled by the URI. [29]

3.1.1 RDF Graph Data Model

The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph. This is illustrated in figure 14, by a node and directed-arc diagram, in which each triple is represented as a node-arc-node link (hence the term "graph"). [30]

Figure 14 - RDF Graph: Subject -> Predicate -> Object [30]

Each triple represents a statement of a relationship between the things denoted by the nodes that it links. Each triple has three parts: [30]

- A Subject
- An Object
- A predicate (also called a property) that denotes a relationship.

The direction of the arc is significant: it always points toward the object. The nodes of an RDF graph are its subjects and objects. The assertion of an RDF triple says that some relationship, indicated by the predicate, holds between the things denoted by subject and object of the triple. The assertion of an RDF graph amounts to asserting all the triples in it, so the meaning of an RDF graph is the conjunction (logical AND) of the statements corresponding to all the triples it contains. [30]

Comparing this to a relational database, to connect two things, one has to add foreign keys to tables, or create join tables, and to link things between databases, one needs ETL jobs like Informatica. In the XML world, connecting things within an XML document is possible, but tedious, and connecting things between XML documents requires XSLT, which involves a lot of work and complexity. The fundamental value and differentiating capability of the Semantic Web is the ability to connect things, URIs make this possible. [30]

3.1.2 RDF Serialization

A number of different serialization formats exist for writing down RDF graphs. However, irrespective of the encoding format of the RDF graphs, they are all logically equivalent. Some of the serialization formats of RDF graphs are: [31]

- Turtle family of RDF languages (N-Triples, Turtle, TriG, and N-Quads)
- JSON-LD
- RDFa
- RDF/XML

The explanation of the serialization formats is out of scope in this thesis. More information about RDF serialization formats can be found at - https://www.w3.org/TR/rdf11-primer

3.2 RDFS – The RDF Schema Language

RDFS is a language with the expressivity to describe the basic notions of commonality and variability familiar from object oriented languages and other class systems, i.e. classes, subclasses, and properties. It provides a data modeling vocabulary for RDF data. RDFS provides mechanisms for describing groups of related resources and the relationships between those resources. RDF Schema is written in RDF defined as resources. These resources are used to determine characteristics of other resources such as domains and range of properties. The RDFS can be broadly using classes and properties.

3.2.1 RDFS Classes

Resources can be divided into groups called classes. The members of a class are known as instances of the class. Classes are themselves also resources. They are identified by URIs and maybe described using RDF properties. RDF distinguishes between a class and the set of its instances. Associated with each class is a set, called the class extension of the class, which is the set of the instances of the class. Two classes may have the same set of instances but be different classes. For example, the tax office may define the class of people living at the same address as the editor of this document. The Post Office may define the class of people whose address has the same zip code as the address of the author. It is possible for these classes to have exactly the same instances, yet to have different properties. Only one of the classes has the property that it was defined by the tax office, and only the other has the property that it was defined by the Post Office. [32]

3.2.2 Properties

RDF property is a relation between subject resources and object resources. The `rdfs:subPropertyOf` property may be used to state that one property is a subproperty of another. If a property P is a subproperty of property P', then all pairs of resources which are related by P are also related by P'. The term super-property is often used as the inverse of subproperty. If a property, P' is a super-property of a property P, then all pairs of resources which are related by P are also related by P'. This specification does not define a top property that is the super-property of all properties. [32]

3.3 OWL – Web Ontology Language

OWL brings the expressivity of logic to semantic web. It allows modelers to express detailed constraints between classes, entities, and properties. OWL was adopted as a recommendation by the W3C in 2004. [27] The current version of OWL, also referred to as “OWL 2”, was developed by the W3C OWL Working Group and published in 2009, with a Second Edition published in 2012. [33]

The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. The Semantic Web will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The first level above RDF required for the Semantic Web is an ontology language what can formally describe the meaning of terminology used in Web documents. If machines are expected to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema. OWL has been designed to meet this need for a Web Ontology Language. [27]

OWL is built on top of RDF. Additional concepts in OWL are implicit in RDF, but it’s too complex to be simply expressed implicitly in RDF.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

3.3.1 OWL Lite

OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. [27] It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies. [27] Owl Lite also has a lower formal complexity than OWL Description Logic, explained in section 3.3.2.

3.3.2 OWL Description Logic

OWL Description Logic (OWL DL) supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). [27] OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL. [27]

3.3.3 OWL Full

OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full. [33]

OWL Lite is the most primitive logic representation, and the complexity goes up as one moves to OWL DL and further in OWL Full. OWL DL is more expressive than propositional logic, but less expressive than First-order logic. [34] The core reasoning problems for DL is usually decidable, and efficient decision procedures have been designed and implemented for these problems. [34] DL is used in artificial intelligence to describe and reason about the relevant concepts of an application domain. [34] OWL Full is potentially NP-hard, meaning, due to its combination of flexibility and expressivity, OWL Full is computationally undecidable with regard to consistency and entailment checking. [35] Therefore, OWL DL has been used to build the ontology model in the Proof of Concept experiments as part of the research effort and the thesis experiment. The power of OWL DL has not been exploited in the PoC experiments though, as it would require integration with an inferencing engine, and that wasn't covered as part of the experiments.

3.4 SPARQL

SPARQL, pronounced 'sparkle', is the standard query language and protocol for Linked Open Data on the web or in a semantic graph database (also called RDF triplestore). SPARQL, short for "SPARQL Protocol and RDF Query Language", enables users to query information from databases or any data source that can be mapped to RDF. The SPARQL standard is designed and endorsed by the W3C and helps users and developers focus on what they would like to know instead of how a database is organized. [36]

Just like SQL allows the user to retrieve and modify data in a relational database, SPARQL provides the same functionality for NoSQL graph databases like GraphDB. In addition, a SPARQL query can also be executed on any database that can be viewed as RDF via middleware. SPARQL is designed to enable Linked Data for the Semantic Web. SPARQL helps grab, massage, create, modify, and delete data. It is the command line access to the underlying RDF graphs. Its goal is to assist people to enrich their data by linking it to other global semantic resources, thus sharing, merging, and reusing data in a more meaningful way. [36]

Correspondingly, a SPARQL query consists of a set of triple patterns in which each element (the subject, predicate and object) can be a variable (wildcard). Solutions to the variables are then found by matching the patterns in the query to triples in the dataset. SPARQL has four types of queries. It can be used to: [36]

- ASK whether there is at least one match of the query pattern in the RDF graph data;
- SELECT all or some of those matches in tabular form (including aggregation, sampling and pagination through OFFSET and LIMIT);
- CONSTRUCT an RDF graph by substituting the variables in those matches in a set of triple templates; or
- DESCRIBE the matches found by constructing a relevant RDF graph.

The biggest strength of SPARQL is navigating relations in RDF graph data through graph pattern matching, where simple patterns can be combined into more complex ones that explore more elaborate relations in the data. Such relations can be explored by using basic patterns, pattern joins, unions, by adding optional patterns that may extend the information about the found solutions, etc.

The wide variety of graph patterns that can be matched through SPARQL queries reflects the wide variety in the data that SPARQL was designed for – the data of the Semantic Web. Whether it is by including optional values so that solutions are not rejected because some part of the pattern doesn't match or by combining graph patterns so that one of several alternatives may match, SPARQL can be used efficiently and effectively to extract the necessary information hidden in non-uniform data stored in various formats and sources. [36]

As the inventor of the World Wide Web, creator and advocate of the Semantic Web and W3C Director, Sir Tim Berners-Lee, puts it: [36]

"Trying to use the Semantic Web without SPARQL is like trying to use a relational database without SQL. SPARQL makes it possible to query information from databases and other diverse sources in the wild, across the Web." – Sir Timothy John Berners-Lee

4. SEMANTIC TECHNOLOGY DECISION SUPPORT SYSTEM – PROOF OF CONCEPT EXPERIMENTS

A lab was setup at MIT Sloan as a hub for proof of concept experiments with the goal of validating the idea that a Semantic Data Lake would be a strong technology candidate for supporting the capability requirements portfolio management, and that the Semantic Data Lake system can augment the decision supporting abilities for the enterprise. A sample of six diverse data sources were used to populate the Semantic Data Lake. The JCEA was used as a driver of the semantic data model. The data models were created as the pillars of the Semantic Data Lake, upon which data would be pushed to, and subsequently retrieve the data back into its original form from the granular structure in the Data Lake. This would validate that the JCIDS and JCEA architecture can be supported in a Semantic Data Lake. The other attempt under the scope of the proof of experiments was to evaluate some of the insights that can be gained, given a Semantic Data Lake with data model and data to support the JCIDS – JCEA. As there was a limited amount of data entered into the Semantic Data Lake, the intent was to provide working examples that demonstrate the range when using such a technology solution, and throw light on the possibilities that are achievable when the Semantic Data Lake is well populated with complete data from varying sources of information. The subsequent parts of this chapter explains the nature of the problem, explains the design of the prototype lab, validating the JCIDS-JCEA architecture on the platform, the procedure used to build the data model as well as upload data into the platform, and finally some of the insights gained from the data contained in the platform.

The enterprise evaluation of JCIDS revealed a sufficient system, capable of maturing into a great system capable of delivering critical insights into key areas for decision maker. [16] The capability requirements portfolio management includes unique, diverse perspectives from multiple stakeholders often give rise to what some describe as a 'wicked problem': [16] *"One that is almost impossible to solve because of the dynamic, contradictory, interrelated, piecemeal decision factors within an environment with inconsistent requirements – and within DoD – sometimes unspoken requirements."* [37]

Given the complex nature of wicked problems, there often is not one unique solution or 'right answer', rather "reasonable multiple solutions". [37] In some cases, the community may not realize that the solution already exists, as it is part of a completely different solution set. [16] So the question then, is how to approach these complex inter related, dynamic problems so they can be effectively managed? [16] One answer is to "break the problem or decision down into more discrete pieces…these 'bite-size' pieces then can be prioritized". [37] However, when resolving these problems down to their 'core', the challenge is to avoid the trap of solving these discrete pieces while ignoring the larger system. [16]

The JCEA semantic data lake is intended as decision support system to a knowledgeable Subject Matter Expert (SME) in analyzing a new capability document undergoing review and validation. The SME is faced with about 2000 prior capability documents stored among 20000 document versions and other materials accessed with great difficulty in KM/DS.

The overall objective is to assist the SME to focus on the important prior documents that cover similar capabilities. The other part is to point the SME to other systems fielded or in development that have similar or related capabilities even if there are no prior JCIDS documents. That is a possible if the Semantic Data Lake is expanded to include information about older systems that did not go through the JCIDS process (which didn't even exist then). The experiments strive to demonstrate the intent of how the platform suits to augment the SME's decision-making process, and not make decisions in lieu of the SME.

The JCIDS ontology design defines a semantic knowledge base that captures the portfolio of capabilities and gaps early in development. The prototype ICD ontology (Appendix C) and the architecture framed the knowledge base. The ontology design also captured and connected essential military and requirements processes, and the subject's domain knowledge. The requirements documents provided the content. These included textual documents interpreted against ontology, structured information in tables and DoDAF artifacts attached in structured form suitable for machine consumption and use, as well as images such as Operational View 1 (OV-1). Additional content came from SME annotations. This is a major step forward in formalizing the JCEA, as data captured and organized in a semantic architecture framework would continue to be accessible and reusable as SMEs rotate in and out as the circumstances change. Appendix C explains the detailed ICD ontology designed as part of previous MIT research work on the JCIDS – JCEA architecture.

The ICD ontology base (see Appendix C) contained 150 data slots based on the draft 2015 JCIDS manual, and the JCEA. The proof of concept experiments was designed to validate the use of a semantic data lake to support the capability requirements portfolio management. The first step was for the semantic data model to be developed to support various constructs of the JCIDS architecture, e.g., JCA, UJT, etc. The next step was to load some of the data that had been extracted from source documents (ICD), and feed them into a semantic web based data lake. There was some intermediate data transformation steps needed to load the data extracted from the source documents into the semantic data store. These are discussed in later sections to follow. The loaded data was interconnected using semantic SPARQL queries to demonstrate how information can be presented with the underlying connected semantic architecture to help create various enterprise viewpoints for different types of stakeholders.

4.1 Design Overview of the Prototype

Standard semantic web technology concepts - OWL, RDF, SPARQL were used to build the proof of concept setup. Figure 15 below shows the architecture of the proof of concept prototype experimental setup.

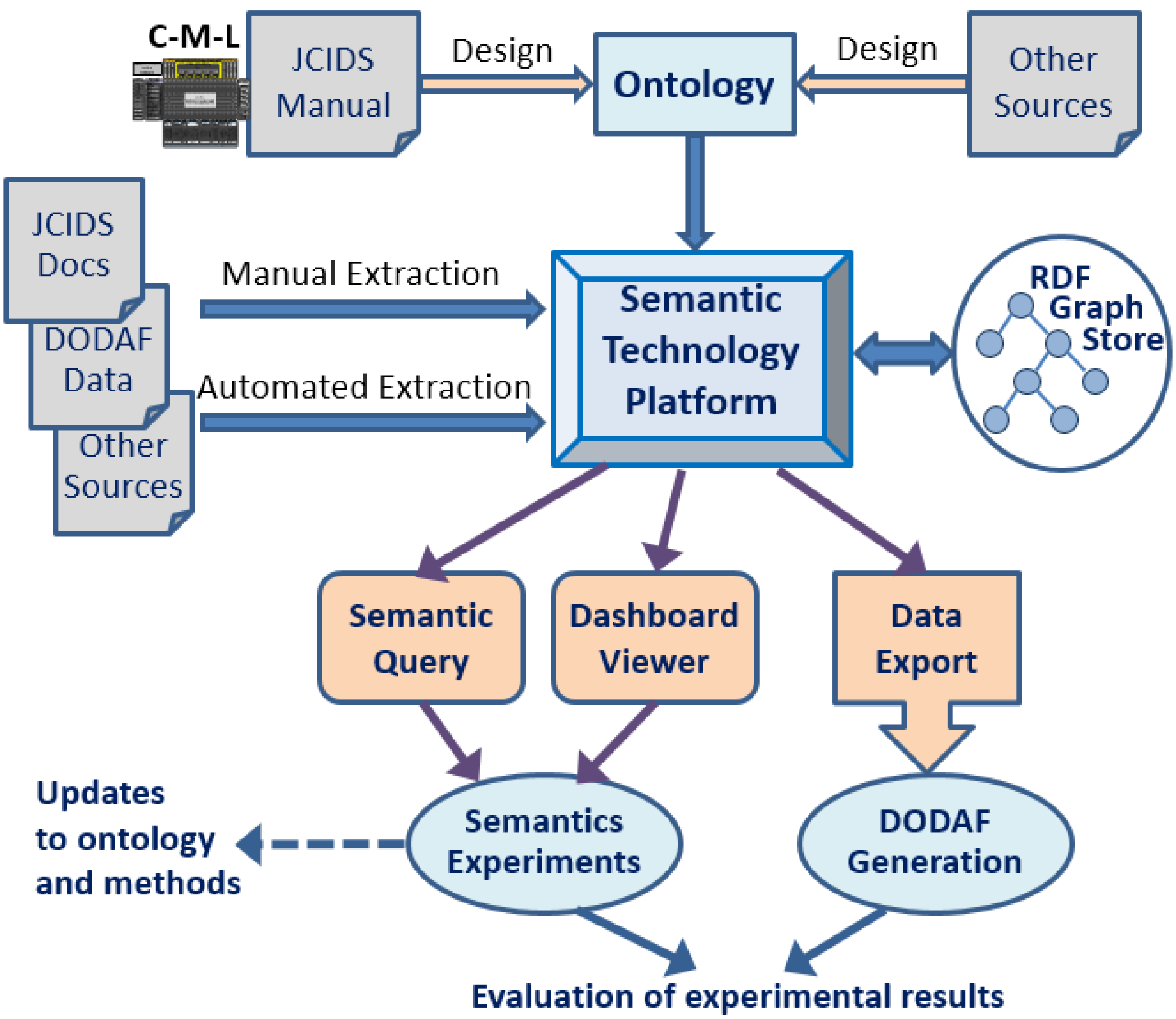


Figure 15 - Semantic Technology Proof-of-Concept Prototype Design Overview [25]

Figure 15 above depicts the architecture of the proof of concept prototype built for the semantic framework decision support system. The JCIDS manual, DoDAF, and JCEA define the ontology that goes into the semantic technology platform. This ontology is the skeleton of the system. The ontology provides the structure to the otherwise unstructured RDF data store. Next, comes the extraction documents. E.g., the ICD extraction documents that were formulated using the ontology concept data slots explained in Appendix C. The data from various sources are then loaded into the semantic technology platform conforming to the canonical form of the ontology design for the semantic technology platform. The semantic data lake is now enriched with both the ontology design and the corresponding data that is living inside the semantic data lake as a RDF graph store. Semantic queries powered by SPARQL language are then triggered into the semantic technology platform. These SPARQL queries reveal the intricacies of the RDF connected triples depending on which object and what the user is intending to know out of the data lake. Based on how the SPARQL queries are formulated, the semantic data lake can be queried and deeper interconnections and similarities between data and objects can be learned over time. The dashboard viewer helps create various viewpoints over the same data lake. The flexibility of the semantic data lake is also that the data can be exported into flat files of various formats, so that it can be consumed by external databases and other systems for cross platform compatibility. The idea is that by performing repeated set of experiments and iterative queries, the semantic technology platform can be fine-tuned to serve the purpose of augmenting as a decision support system. The intent is that, later, a user-friendly web interface can be built to trigger known SPARQL queries to support critical decisions, e.g. What are the dependencies that a system like the JCM/JAGM has with other systems that are currently fielded. Ability to provide a quick turnaround to such questions can greatly enhance the decision making speed and ability of the organization as a whole.

4.2 Semantic Data Lake Design and Build

This section validates the JCIDS architecture framework on the semantic technology data platform. This architecture can provide valuable insights to DoD leadership on the system, capability, and mission dependencies between individual systems within the enterprise. The semantic technology platform used here is the Anzo Smart Data Lake, a product from Cambridge Semantics under a non-production evaluation license. (https://www.cambridgesemantics.com/)

Figure 16 shows the ontology build, load, and data upload into the semantic data lake platform – Anzo.

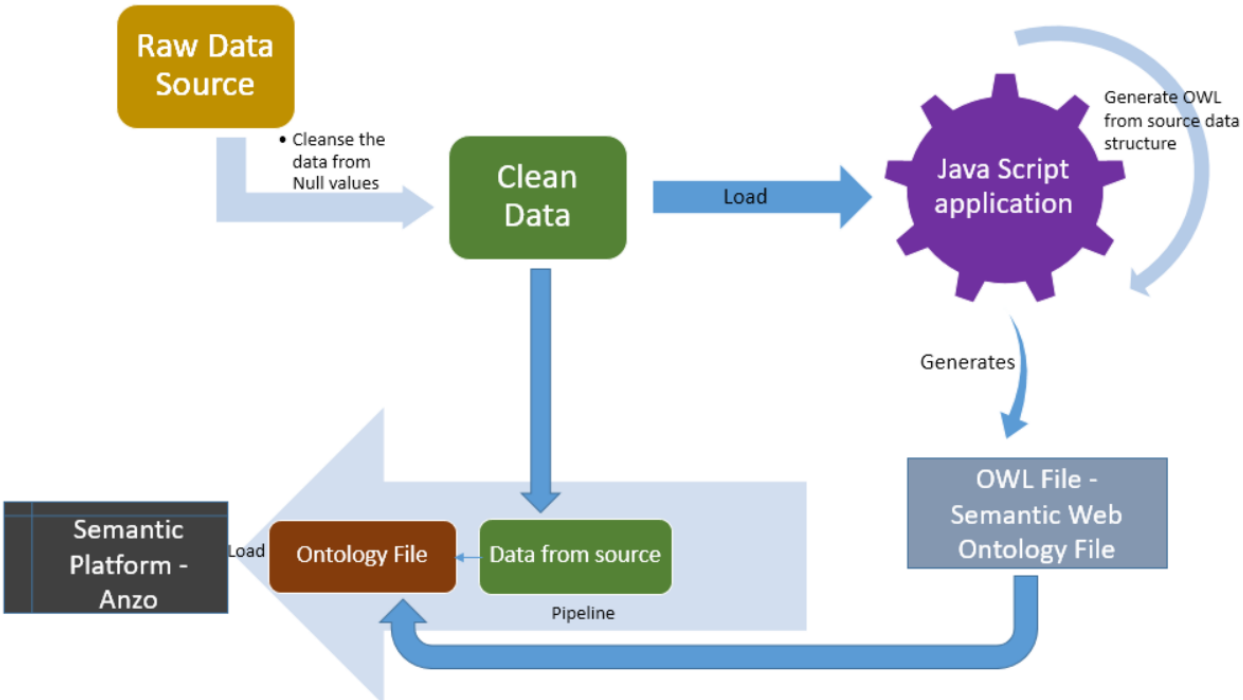


Figure 16 - High-Level steps of the implementation process

The generic implementation process included the following high-level steps –

- The raw data, which comes from the ICD ontology extraction documents (Appendix C), was first cleansed to align the data with the headers and remove any null or empty cells for further pre-processing.
- Each document type of document, e.g., ICD ontology, JCA, and other ontology files in excel were loaded into an OWL generation software coded in JavaScript and Node.js server.
- The data was then used in a Java Script application built to generate OWL files corresponding to the data structure being passed to it. Anzo Smart Data Lake provides a graphical user interface to build business ontology manually. However, an intermediate Java Script application was written to automate the OWL file generation, which can greatly speed up the ontology creation for large ontologies.
- The OWL file is imported into the Anzo semantic platform to store the ontology in the Anzo software.
- The data corresponding to each ontology is uploaded into the Anzo platform for analytics.

4.2.1 Semantic Transformation Architecture of the Ontology Documents

The ontology existed only implicitly in documents before converting them into the semantic web OWL. Figure 17 shows the semantic transformation of the various types of ontology documents. The ontology documents can be the structure of various sources of information, e.g., the ICD Ontology documents, JCA Taxonomy, UJT List, etc. These are covered in details in section 4.3.

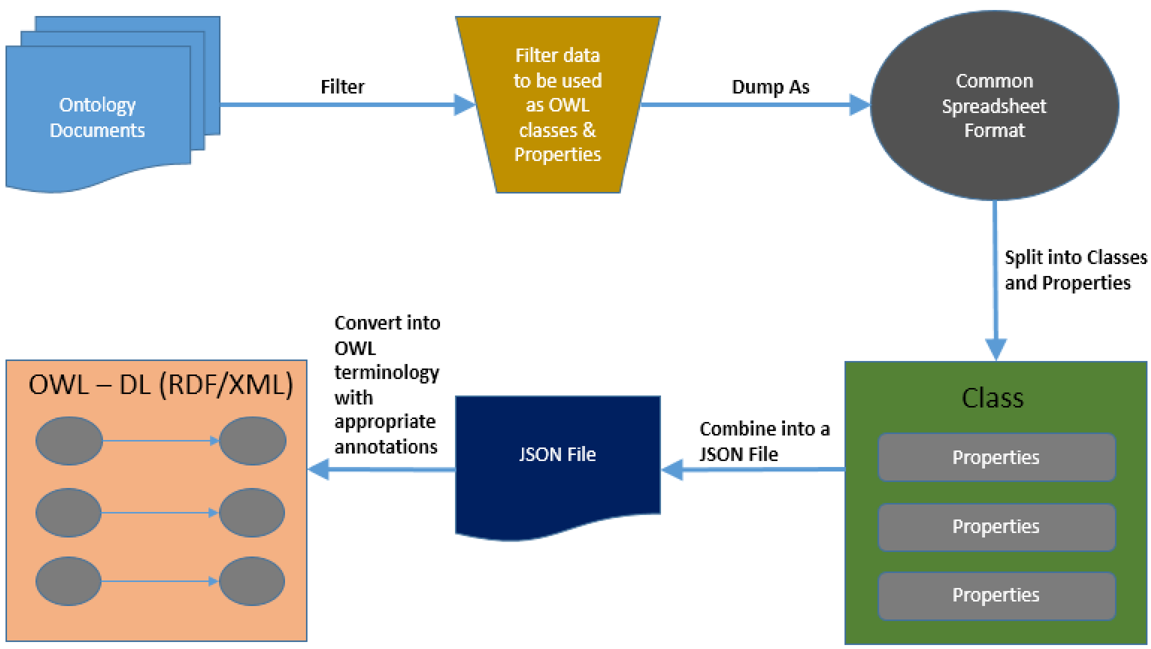


Figure 17 - Semantic Transformation of ontology documents to OWL

The ontology documents, as described in Appendix C, were processed via the OWL generation architecture as shown in figure 16. The OWL file generation was performed with the workflow as described below –

- The Excel documents with implicit ontology were filtered manually to pull the ontology structures out.
- The JavaScript application then splits the ontology data (classes, properties, description, etc.) into a common spreadsheet format in the main memory for pre-processing.

- These are then combined into an intermediate JSON file that holds the raw ontology information. Figure 18 shows a snippet of the JSON extract of the ICD ontology. The JSON data element is an array, that describes the name of the data element (Feature), type of the data whether it is a class or property, etc. (Type), base classes if any (Base_Class), description, domain of the property (class that owns the property/triple), range (class that the property/triple maps to), and the mathematical property (characteristics).
- The application then converts the JSON file into OWL (semantic ontology). During the conversion, the intrinsic relationships are injected into the OWL transformations. This includes relationships such as hierarchy of classes, domain, range, and characteristics of the properties (functional, transitive, etc), shown in figure 19.

```
{
  "name": "ICD_Ontology",
  "data": [
    [
      "Feature",
      "Type",
      "Base_Class",
      "Description",
      "Domain",
      "Range",
      "Characteristics"
    ],
    [
      "ReferenceData",
      "Class",
      "None",
      "Program or Package information"
    ],
    [
      "CoverPage",
      "Class",
      "None",
      "Information retrieved from cover page of the ICD"
    ],
    [
      "ThreatContext",
      "Class",
      "None",
      "Contextual information about the threat environment"
    ]
    [
      "Program_Name",
      "Property",
      "Name of the program or package as mapped to",
      "ReferenceData",
      "String",
      "Functional"
    ]
  ]
}
```


Figure 18 - Snippet of the JSON data generated from the ICD Ontology

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:anzoowl="http://openanzo.org/ontologies/2009/05/AnzoOwl#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <owl:Ontology rdf:about="http://sloan.mit.edu/ontologies/ICD_Ontology">
        <dc:title>ICD_Ontology</dc:title>
        <rdfs:label>ICD_Ontology</rdfs:label>
    </owl:Ontology>
    <owl:Class rdf:about="http://sloan.mit.edu/ontologies/ICD_Ontology#ReferenceData">
        <dc:title>ReferenceData</dc:title>
        <rdfs:label>ReferenceData</rdfs:label>
    </owl:Class>
    <owl:Class rdf:about="http://sloan.mit.edu/ontologies/ICD_Ontology#CoverPage">
        <dc:title>CoverPage</dc:title>
        <rdfs:label>CoverPage</rdfs:label>
    </owl:Class>
    <owl:Class rdf:about="http://sloan.mit.edu/ontologies/ICD_Ontology#ThreatContext">
        <dc:title>ThreatContext</dc:title>
        <rdfs:label>ThreatContext</rdfs:label>
    </owl:Class>
    <rdf:Property rdf:about="http://sloan.mit.edu/ontologies/ICD_Ontology/ReferenceData#Program_Name">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
        <OntologyCluster rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        xmlns="http://sloan.mit.edu/ontologies/ICD_Ontology#">1</OntologyCluster>
        <OntologySlotTag rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        xmlns="http://sloan.mit.edu/ontologies/ICD_Ontology#">102</OntologySlotTag>
        <dc:title>Program_Name</dc:title>
        <rdfs:domain rdf:resource="http://sloan.mit.edu/ontologies/ICD_Ontology#ReferenceData"/>
        <rdfs:label>Program_Name</rdfs:label>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    </rdf:Property>
</rdf:RDF>
```

Figure 19 - Snippet of the OWL document generated from the JSON for the ICD Ontology

This conversion of the ontology documents into OWL demonstrates that the JCIDS – JCEA architecture can be codified into a workflow to be able to generate semantic ontology to be processed by an engine. Workflows can be created, modified, and destroyed by end users (SMEs) with a graphical user interface, to pre-process various JCIDS, and other architectural, conceptual, and capability documents. The detailed results of the semantic ontology creation from various JCIDS concepts have been described in section 4.3.

4.2.2 Ontology Mapping in the Semantic Data Lake

As described in the start of this chapter 4.2, Cambridge Semantics data-lake software platform Anzo stored the ontology for the analytics. This ontology was created from the OWL files generated, as described in section 4.2.1.

The ontology files mapped into the Anzo Semantic Data Lake platform now hosts the structure of the RDF data that can be uploaded against it. Figure 20 shows a snippet of the ontology classes with the hierarchy mapped into the Anzo Semantic Data Lake platform. This demonstrates that conceptual and implicit ontologies residing in various types of architecture, capabilities and operational documents can be built into a standard out of the box semantic data lake platform for analysis.

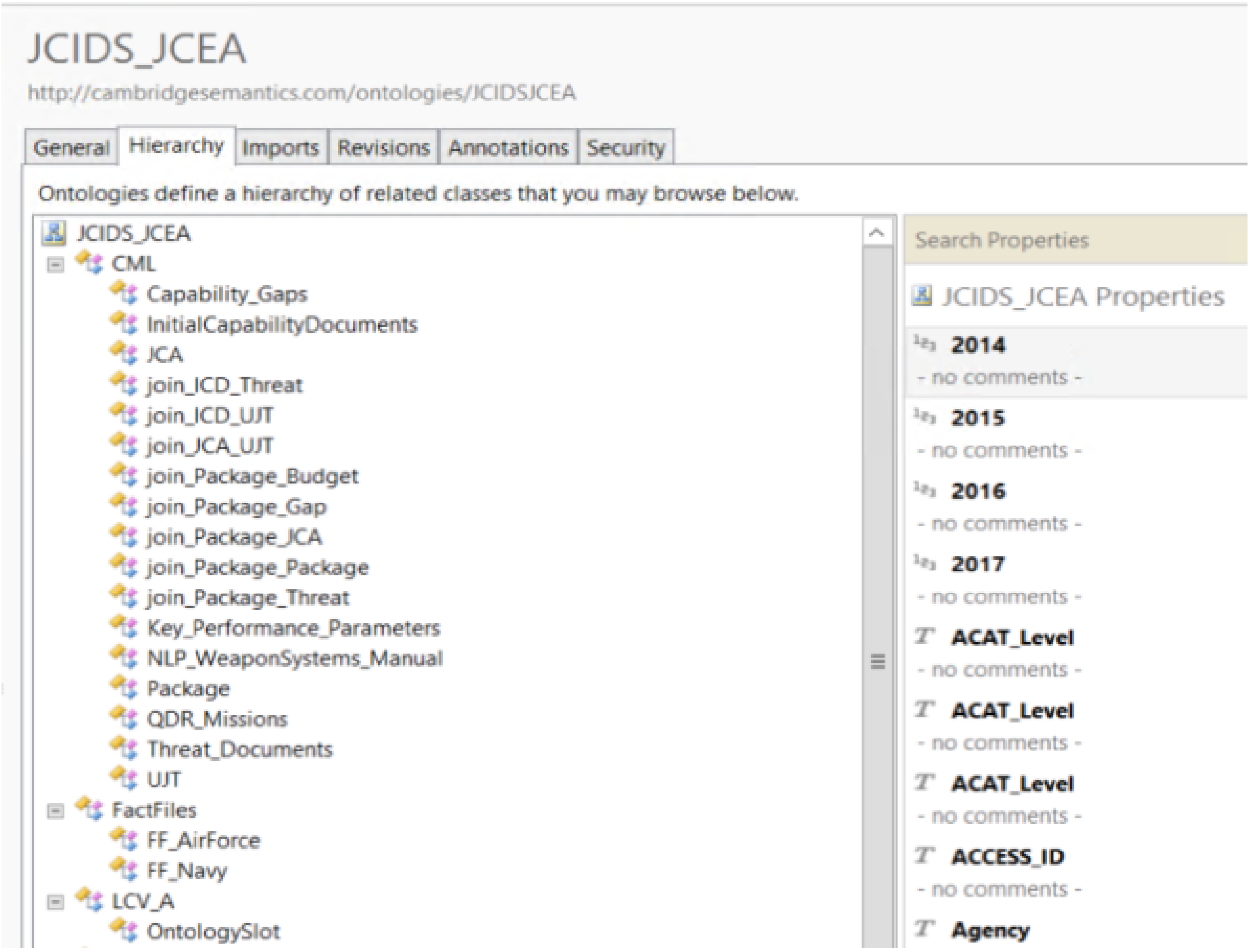


Figure 20 - Ontology Mapping created in Anzo Semantic Data Lake

4.3 JCIDS Architecture validation on Semantic Data Lake Platform

Let's consider the JCM-JAGM case as discussed in section 1.1. The SME who takes up work on a reinstated program like the JAGM program should be able to identify prior editions of JAGM and its predecessor JCM. The JCM-JAGM is a classic case of evolving requirements with changing budgets, threats, priorities, and other competing solutions. The SME should be alerted to the six competing HELLFIRE system variants that provide capabilities similar to the JCM-JAGM. In addition, the SME should be able to identify the SDB (Small Diameter Bomb, e.g., GBU-39) as an alternative precision guided munition capability. All these capabilities perform the same task, although under different conditions and with different features. The other piece of the puzzle here is the integration problem. When a new missile like JAGM is connected to an aircraft, the pilot should be able to use the missile with the interface that he/she used with the previous HELLFIRE versions. This is where the SME should also be able to see the related aircraft that each is (or is intended to be) integrated with. The SME should be able to identify the dependencies that one need to be aware of, or maybe changes required to the interfaces of either the missile, or the aircraft, or the human machine interface of the pilot's cockpit for the missile integration.

The JCIDS defined in a JCEA provides an enterprise perspective of the different systems that interact to provide value to the warfighters. This enterprise perspective becomes important as decision makers must gain an understanding of the "as-is" architecture and the impacts a proposed system has on the enterprise. [3] The various JCIDS Architecture concepts validated on the Anzo Semantic Data Lake platform are described in the following sub-sections.

4.3.1 Joint Capability Areas (JCA)

As discussed briefly in Section 2.5, the DoD utilizes JCAs as a common language to discuss and describe capabilities across all of DoD requirements. A sample of three of the nine JCAs, with their associated Tier 2 and Tier 3 elements is shown in Table 3 below.

JCA Tier 1	JCA Tier 2	JCA Tier 3
Force Support	Force Management	Global Force Management
		Force Configuration
		Global Posture Execution
		Readiness Reporting
	Force Preparation	Training
		Exercising
		Educating
		Doctrine
		Lessons Learned
		Concepts
		Experimentation
Battlespace Awareness	Planning & Direction	Define and Prioritize Requirements (P&D)
		Develop Strategies (P&D)
		Task and Monitor Resources (P&D)
	Collection	Signals Collection
		Imagery Collection
		Measurements and Signatures Collection
		Human Based Collection
Force Application	Maneuver	Maneuver to Engage (MTE)
		Maneuver to Insert (MTI)
		Maneuver to Influence (MTInfl)
		Maneuver to Secure (MTS)
	Engagement	Kinetic Means
		Non-Kinetic Means

Table 3 - Sample JCA Tiers 1 – 3

```
{
 "name": "Sheet1",
 "data": [
  [
   "JCA_ID",
   "JCA_Name",
   "Base_Class",
   "JCA_Definition"
  ],
  [
   1,
   "Force Support",
   0
  ],
  [
   1.1,
   "Force Management",
   "1",
   "The ability to integrate new and existing human and technical
   assets from across the Joint Force and its mission partners to
   make the right capabilities available at the right time and place
   to support National security."
  ],
  [
   "1.1.1",
   "Global Force Management",
   "1.1",
   "The ability to align force apportionment, assignment, and
   allocation methodologies in support of the National Defense
   Strategy and joint force availability requirements; present
   comprehensive insights into the global availability and
   operational readiness"
  ]
  [
   "3.2.1",
   "Kinetic Means",
   "3.2"
  ]
  ]
}
```


Figure 21 - JSON expression of the JCA

Figure 21 shows the intermediate JSON expression of the JCA defined in the JCA documents. Notice that the whole data is compartmentalized into JCA id, JCA Name, JCA Definition (Description of the JCA), and Base Class. The base class points to the parent JCA id, if there exists any. E.g., JCA id 1.1 has a base class of JCA id 1, but JCA id 1 has no base class, since it is one of the roots of the JCA hierarchy.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:anzoowl="http://openanzo.org/ontologies/2009/05/AnzoOwl#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <owl:Ontology rdf:about="http://sloan.mit.edu/ontologies/JCA">
      <dc:title>JCA</dc:title>
      <rdfs:label>JCA</rdfs:label>
   </owl:Ontology>
   <owl:Class rdf:about="http://sloan.mit.edu/ontologies/JCA#1">
      <dc:title>1</dc:title>
      <rdfs:label>1</rdfs:label>
   </owl:Class>
   <owl:Class rdf:about="http://sloan.mit.edu/ontologies/JCA#1.1">
      <dc:title>1.1</dc:title>
      <rdfs:label>1.1</rdfs:label>
      <rdfs:subClassOf rdf:resource="http://sloan.mit.edu/ontologies/JCA#1"/>
   </owl:Class>
   <owl:Class rdf:about="http://sloan.mit.edu/ontologies/JCA#1.1.1">
      <dc:title>1.1.1</dc:title>
      <rdfs:label>1.1.1</rdfs:label>
      <rdfs:subClassOf rdf:resource="http://sloan.mit.edu/ontologies/JCA#1.1"/>
   </owl:Class>
   <owl:Class rdf:about="http://sloan.mit.edu/ontologies/JCA#1.3.2.1">
      <dc:title>1.3.2.1</dc:title>
      <rdfs:label>1.3.2.1</rdfs:label>
      <rdfs:subClassOf rdf:resource="http://sloan.mit.edu/ontologies/JCA#1.3.2"/>
   </owl:Class>
<rdf:Property rdf:about="http://sloan.mit.edu/ontologies/JCA/CapabilityAreaName">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
   <dc:description>Joint Capability Area Name</dc:description>
   <dc:title>CapabilityAreaName</dc:title>
   <rdfs:comment>Joint Capability Area Name</rdfs:comment>
   <rdfs:domain rdf:nodeID="0b367612-80e8-4a94-b3d4-4caf0c9dc98c"/>
   <rdfs:label>CapabilityAreaName</rdfs:label>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Property>
```

Figure 22 - OWL snippet of the JCA ontology

Figure 22 shows a snippet of the JCA ontology conversion into OWL expression that was built using the procedure covered in section 4.2.1. Everything is defined as a URI resource here, and this is the key – the ability to interconnect and interlink each data fragment (JCA id, or JCA Name in this case) with other data fragments, such as programs, capabilities, requirements etc. This provides freedom to the JCA concepts to be a connecting factor, and viewing various components via the joint capability areas lens. E.g., the JAGM program can be looked up to check what JCAs are connected to it and a reverse lookup can be designed to find out what other systems share the same JCAs. This is covered in more details in section 4.4.

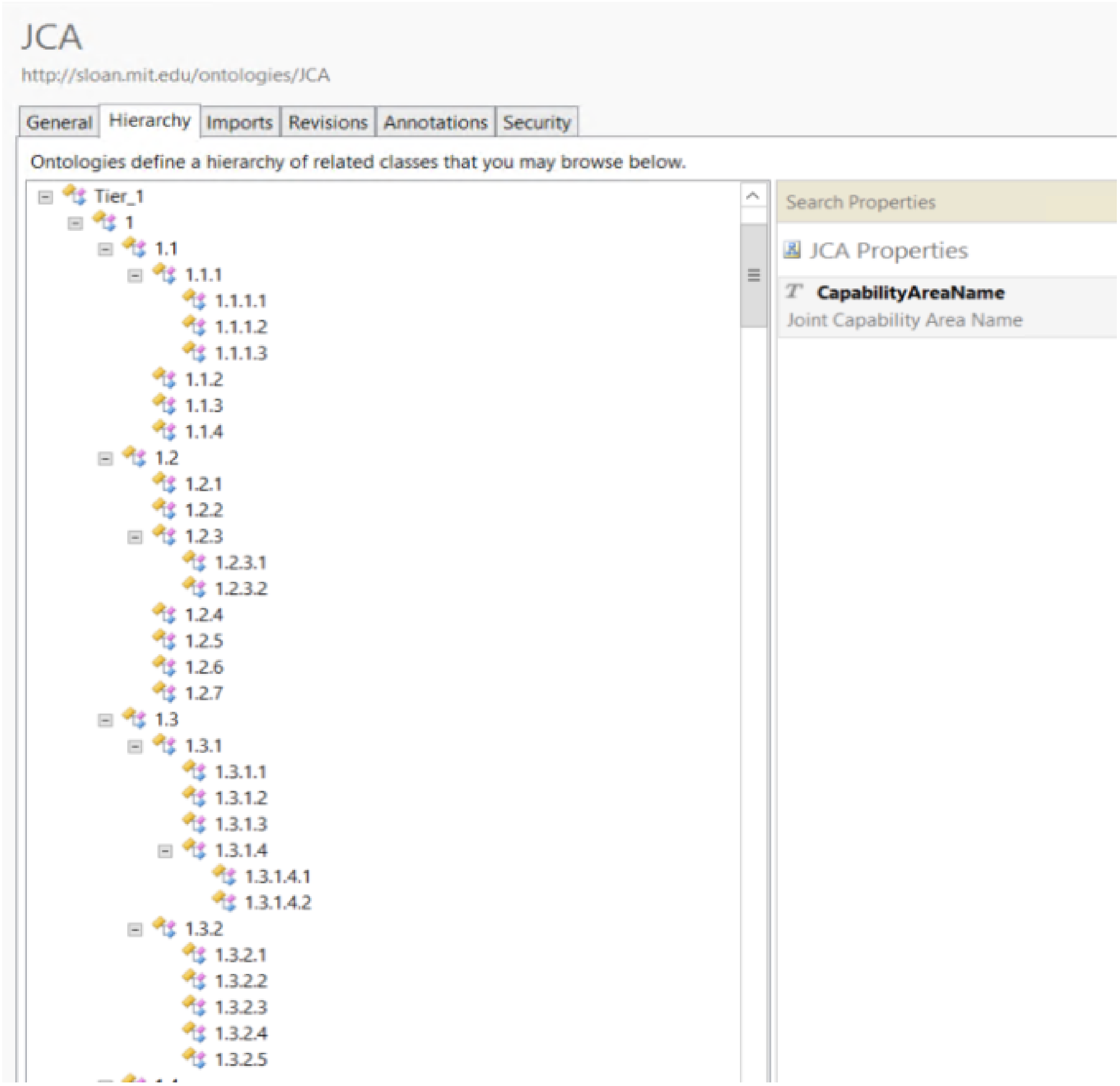


Figure 23 – Snippet of JCA (Joint Capability Area) Ontology

The excel documents that contain detailed coverage of the various JCA is processed through the transformations, via the steps shown in figure 21 and figure 22 using the procedure as described in section 4.2.1 and section 4.2.2, to create semantic representations of the capability areas and their hierarchy. Figure 23 shows the JCA ontology built when ported into the Semantic Data Lake platform. This results in the JCA to be codified and connected to any capability information that may be loaded into the Semantic Data Lake platform. The capability requirements lived inside JCIDS documents, and the JCA were embedded deep inside the documents. Creating the JCA inside the Semantic Data Lake platform now gives the potential to connect capabilities, requirements, and existing systems under the viewpoint of JCA commonality.

4.3.2 Universal Joint Task List (UJT) and References

UJTs are the missions of the warfighters, grouped together hierarchically for coherency in the capabilities' converging together to accomplish a certain high-level mission. The material capabilities realized via different systems need to be fielded to achieve military missions. These missions are categorized and assembled together into the concept of UJTL under the JCIDS architecture. A sample of four different types of mission areas are shown in Table 4.

Category	Task Number	Task Title	References
Strategic National	SN 1	Conduct Deployment and Redeployment	JP 3-35 (primary), JP 4-09
Strategic National	SN 1.1	Determine Transportation Infrastructure	JP 3-35, JP 4-01 (primary), JP 4-09, CJCSM 3122 Series, CJCSM 3130.03
Strategic National	SN 1.1.1	Determine Transportation Support	JP 3-17, JP 4-01 (primary), JP 4-01.2, JP 4-01.5
Strategic National	SN 2	Provide Strategic Intelligence	JP 2-0, JP 2-01 (primary), JP 2-01.3, Title 50 U.S.C
Strategic National	SN 2.1	Manage Strategic Intelligence	JP 2-0, JP 2-01 (primary), JP 2-01.3
Strategic National	SN 2.1.1	Develop National Strategic Intelligence Policy	JP 2-0 (primary)
Strategic Theater	ST 1	Deploy Forces	JP 3-35 (primary), JP 4-0, JP 5-0
Strategic Theater	ST 1.1	Conduct Intratheater Deployment	JP 3-33, JP 3-35 (primary), JP 4-01, CJCSM 3130.03
Strategic Theater	ST 1.1.1	Process Requests for Forces/ Requests for Capabilities (RFF/ RFC)	JP 4-09 (primary), CJCSM 3122 Series
Operational	OP 1	Conduct Operational Maneuver	JP 3-0, JP 3-02, JP 3-06, JP 3-09, JP 3-31 (primary), JP 3-32
Operational	OP 1.1	Conduct Operational Movement	JP 4-0, JP 4-01, JP 4-09 (primary)
Operational	OP 1.1.1	Formulate Deployment Request	JP 3-35 (primary), CJCSM 3122 Series, CJCSM 3130.03
Tactical	TA 6	Protect the Force	JP 3-0, JP 3-07.2, JP 3-07.3, JP 3-10 (primary), JP 3-11
Tactical	TA 6.1	Provide Explosive Ordnance Disposal (EOD) Support	JP 3-15.1, JP 3-28, JP 3-34, JP 3-42 (Primary)
Tactical	TA 6.10	Prevent Improvised Explosive Devices (IEDs)	JP 3-15.1 (primary), JP 3-28, JP 3-34, JP 3-42

Table 4 - Sample of UJT under four different categories

The documents for the UJT contained the official UJT task number, the task description, and the corresponding references that the mission tasks has to the Joint Chiefs of Staff library, doctrines, publications, and other directorates. This is a critical link for the data extracted from documents and loaded into the Semantic Data Lake platform. It helps connect the references used in the documents (ICD, CDD, etc.) to the UJTs that can apply to the capabilities. Figure 24 shows the hierarchy of the UJT classes. The four levels of war – strategic national, strategic theater, operational, and tactical are defined as the child classes under the Universal Joint Task base class (depicted with dotted lines in figure 24).

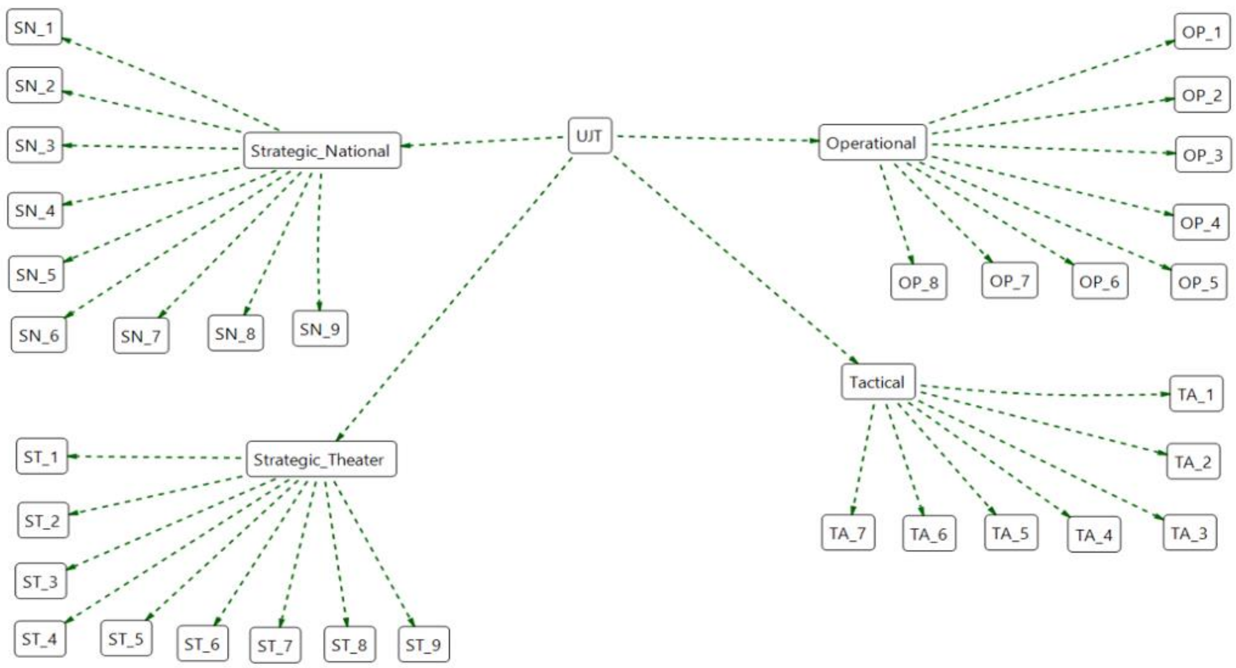


Figure 24 - Semantic class hierarchy and linking of the UJT to the References

Snippet of the OWL representation of the UJT ontology is shown in figure 25. Figure 24 shows the top three levels of hierarchy of the UJT classes in the Anzo Semantic Data Lake ontology tool. The UJT ontology is a basic high-level ontology that only defines the types of tasks can reside inside the Semantic Data Lake. The choice of design of the UJT ontology depends upon how deep an analysis an SME wants to perform. For the scope of this PoC, the ontology design was chosen to be a high-level one, to demonstrate how the concepts of missions can be imbibed into the Semantic Data Lake platform. The final design choice is upto the discretion of the user, but it is recommended to go more granular in defining the ontology for the concept. However, there is a tradeoff on the flexibility as more granular ontology design is chosen for the concepts.

```xml
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:anzoowl="http://openanzo.org/ontologies/2009/05/AnzoOwl#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <owl:Ontology rdf:about="http://cambridgesemantics.com/ontologies/UJTLRefs">
      <dc:title>UJTL_Refs</dc:title>
      <rdfs:label>UJTL_Refs</rdfs:label>
   </owl:Ontology>
   <owl:Class rdf:about="http://cambridgesemantics.com/ontologies/UJTLRefs#Operational">
      <dc:title>Operational</dc:title>
      <rdfs:label>Operational</rdfs:label>
      <rdfs:subClassOf rdf:resource="http://cambridgesemantics.com/ontologies/UJTLRefs#UJTL" />
   </owl:Class>
   <rdf:Property rdf:about="http://cambridgesemantics.com/ontologies/UJTLRefs#references">
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
      <dc:title>References</dc:title>
      <rdfs:domain rdf:nodeID="udbb02213-1af2-4eb9-8ea6-1caa82838be4" />
      <rdfs:label>References</rdfs:label>
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
   </rdf:Property>
   <owl:Class rdf:about="http://cambridgesemantics.com/ontologies/UJTLRefs#References">
      <dc:title>References</dc:title>
      <rdfs:label>References</rdfs:label>
   </owl:Class>
   <rdf:Property rdf:about="http://cambridgesemantics.com/ontologies/UJTLRefs#refers">
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty" />
      <dc:title>refers</dc:title>
      <rdfs:domain rdf:nodeID="ud5ad7aa2-83a4-4e93-a940-fbb973eda0b8" />
      <rdfs:label>refers</rdfs:label>
      <rdfs:range rdf:resource="http://cambridgesemantics.com/ontologies/UJTLRefs#References" />
   </rdf:Property>
</rdf:RDF>
```

Figure 25 – Snippet of the OWL representation of UJT

4.3.3 Capability Gaps and Capability Gap Overlaps

Capability gaps are the drivers behind the rise of capability requirements. When the current set of fielded solutions has operational gaps or capability gaps, they need to be filled to close the current gap. The ICD specifies a capability requirement, derived from warfighter needs, and identifies any gaps in capability based on current systems. The CDD provides additional detail on the capability requirement and specifies key performance parameters that potential solution must achieve to close the capability gap identified in the ICD. Finally, the CPD specifies capability requirements that support the production of a single material solution to close the gap.

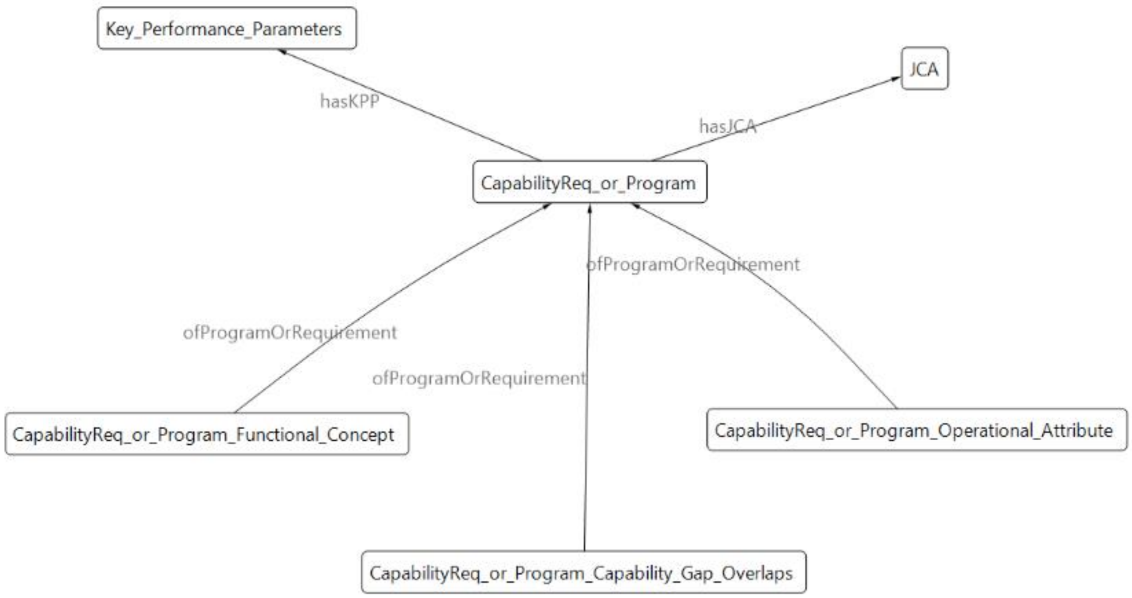


Figure 26 - Capability Gaps and Capability Gap Overlaps ontology as viewed in Anzo Semantic Data Lake Platform

Figure 26 shows the ontology fragment of a capability requirement or program (class) and its relationships. The capability gaps and overlaps, operational attributes, and functional concepts in figure 26 have an "ofProgramOrRequirement" property connecting to a specific capability requirement or program. The capability requirement or program has a "hasJCA" relationship with the JCA class (figure 26). The capability requirement or program class also has "hasKPP" relationship with the key performance parameters class (figure 26). E.g., The JAGM program would be an instance under the class capability requirement or program; Force Application would be a functional concept that applies to the JAGM program. Self-guided missile would be an example of a capability gap of the JAGM program. Similarly, other relationships can be ingrained into the data that is fed into the Semantic Data Lake platform, in accordance with the ontology design it is uploaded to.

4.3.4 JCEA - ICD Ontology

ICD Ontology as discussed in details in Appendix C, provides the constructs to granulize the data about capability requirements and programs into fragments of data, and interlink the connections to preserve the semantics.

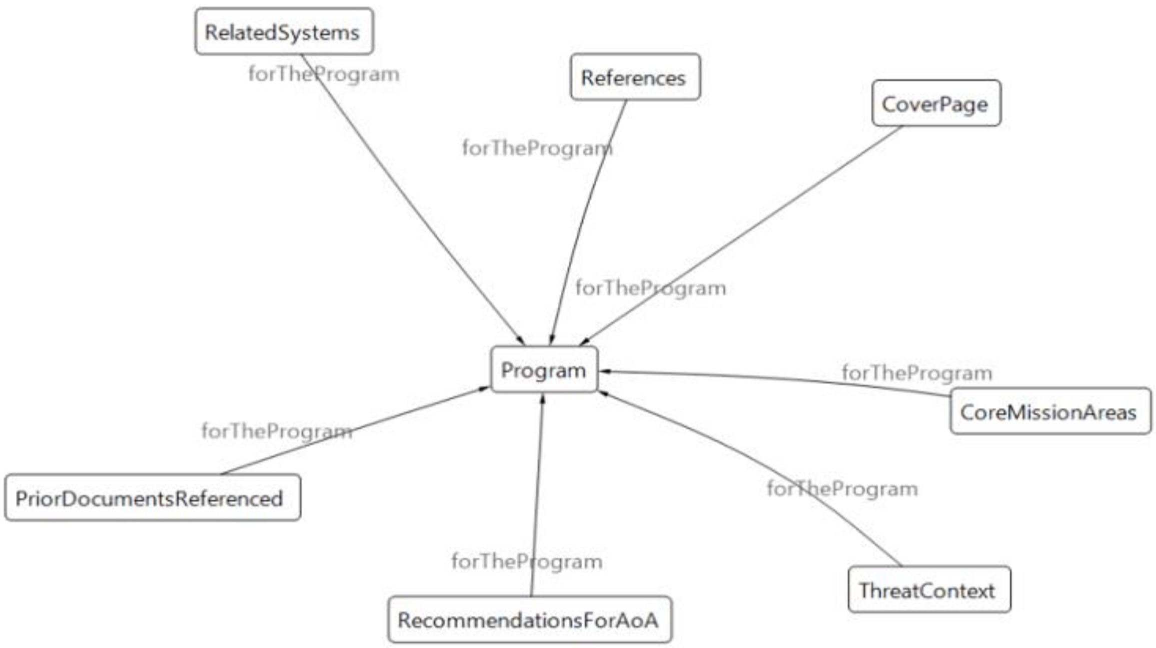


Figure 27 - ICD Ontology as viewed in Semantic Data Lake Platform

Figure 27 shows the ICD ontology snippet as viewed in the Anzo Semantic Data Lake Platform. The Reference Data class has the information about the capability requirement document (ICD). The other characteristics of a capability requirement program such as references in an ICD, recommendations for AoA (Analysis of Alternatives), Core Mission Areas, Threat Context, etc. are connected to a program or requirements via the "forTheProgram" contextual link. This holds an ICD ontology together into a set of connected data features.

4.3.5 Overview of results from the JCIDS Architecture Validation on Semantic Data Lake Platform

Before building the ontology on a Semantic Data Lake Platform, all the ontology structures existed inside excel documents. Most prominently, the collection of ICD ontology

(Appendix C) documents had the ontology constructs broken down by ontology slots and ontology slot tags, and data fragments were defined against each ontology slot. These documents thus had the semantics (ontology, data, and their relationships), all defined in a two dimensional format, making it extremely difficult for an analyst to compare features of documents and thereby compare and contrast systems, requirements, and capabilities.

Splitting apart the ontology constructs into semantic OWL files helped extract the canonical form of the data contained in the excel documents. Then, the data uploaded against the ontology populated the Anzo platform, which created the "semantic data lake". Data in the Semantic Data Lake existed in RDF triples, a true semantic technology representation. The following chapter 4.4 describes how the data was uploaded into the Anzo platform and what were the key insights gained from the semantic data lake platform that can aid in decision making for the SME analyst.

4.4 Uploading Data into the Anzo Platform and Visualization

Anzo software provides tools and interfaces to interact with the Semantic Data Lake platform. The tool to upload data into the Anzo platform was a Microsoft Excel add-in called "Anzo for Excel", that provides ability to map data, contained in excel with the ontologies defined in the platform.

4.4.1 Data Upload Procedure on the Anzo Platform

The interface to perform the work is a linked data set, which contains a set of ontologies put under a logical grouping. The user of the software can also define the data source (internal graph database, managed by the Anzo software) to be used for the data uploads.

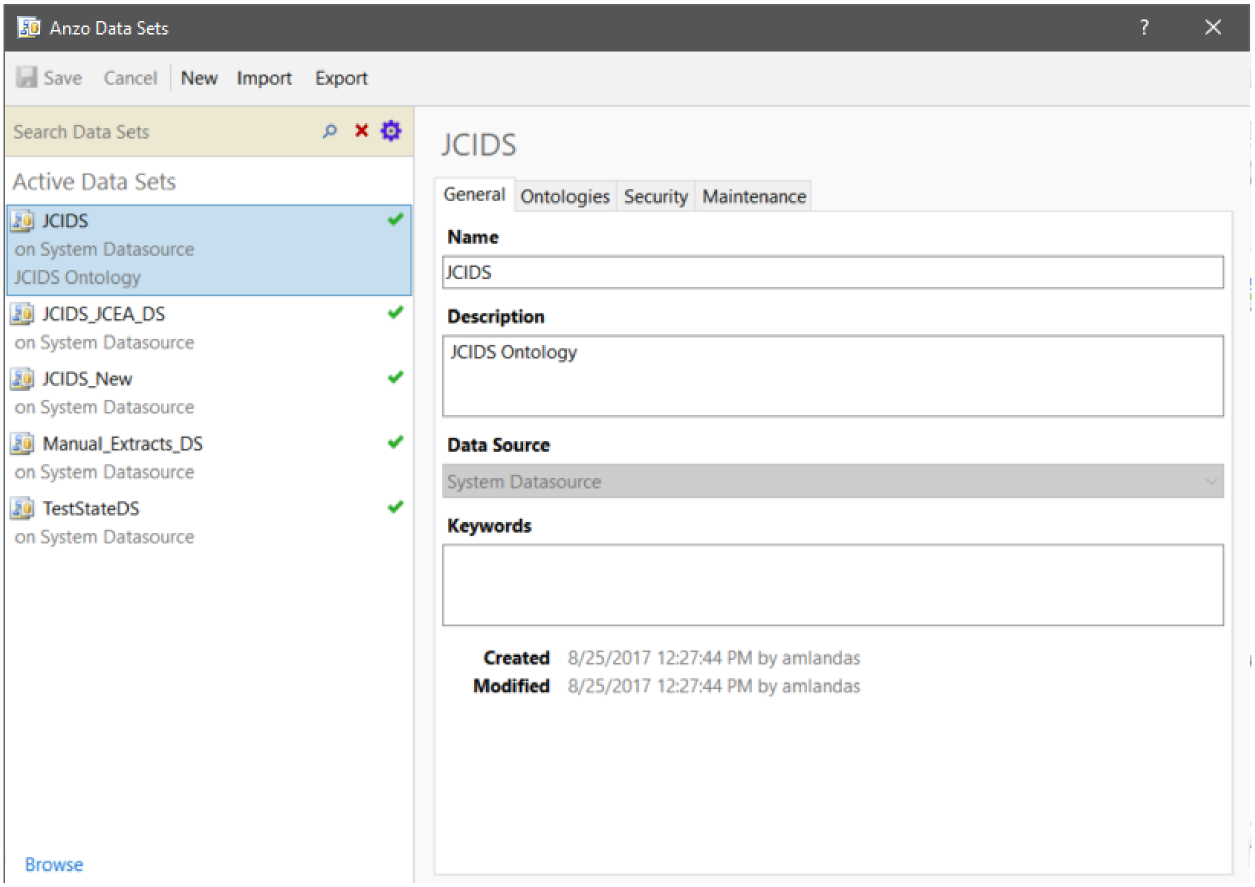


Figure 28 - Defining Data sets in "Anzo for Excel" tool, in the Semantic Data Lake platform against selected ontology for data upload

Figure 28 shows the interface to define data sets in the "Anzo for Excel" tool of the Anzo Semantic Data Lake software. These data sets are the ontology groups under which the data in the excel sheets would be uploaded. One of the requirements when using the Anzo tool to upload data was to flatten out the data contained in the ontology files. E.g., The ICD ontology documents had the ontology definition and the corresponding data adjacent to each other in the excel data format (Table 5 shows the original structure of the ICD ontology data). The data fields had to be transposed from vertical layout, into a horizontal layout (shown in Table 6), to upload the data into the Semantic Data Lake platform.

Extracted Text or Data	Ontology Concept Data Slot Description	Seq Num	Ontology Cluster #	Ontology Slot Tag
177	ACCESS ID (if known from prior analysis)		1	101
LCV	Program Name		1	102
Land	JRAD Portfolio		1	103
457	PNO		1	104
0	Program Cost		1	105
Combat Vehicle	Commodity Type		1	106
2017	IOC (actual intended or assumed for planning)		1	107
2040	FOC (actual intended or assumed for planning)		1	108
	Retirement (assumed or planned)		1	109

Table 5 - Snippet of structure of ICD ontology for a Land based Combat Vehicle Program

ACCESS ID	Program Name	JRAD Portfolio	PNO	Program Cost	Commodity Type	IOC	FOC	Retirement
177	LCV	Land	457	0	Combat Vehicle	2017	2040	

Table 6 - Snippet of the ICD ontology for a Land Based Combat Vehicle program, transposed to upload data into Anzo Semantic Data Lake

4.4.2 Visualizing the Data Uploaded into the Anzo Semantic Data Lake platform

“Anzo on the Web” product gives the ability to visualize the data uploaded into the Anzo Data Lake platform against the ontologies. The “Anzo on the Web” provides a web-based interface to the Semantic Data Lake platform. This web-based interface can be used to view the existing data in the platform, and also to query and analyze the data using SPARQL queries. Figure 29 shows the visualization of the land based combat vehicle program as viewed in the Anzo on the Web visualizer.

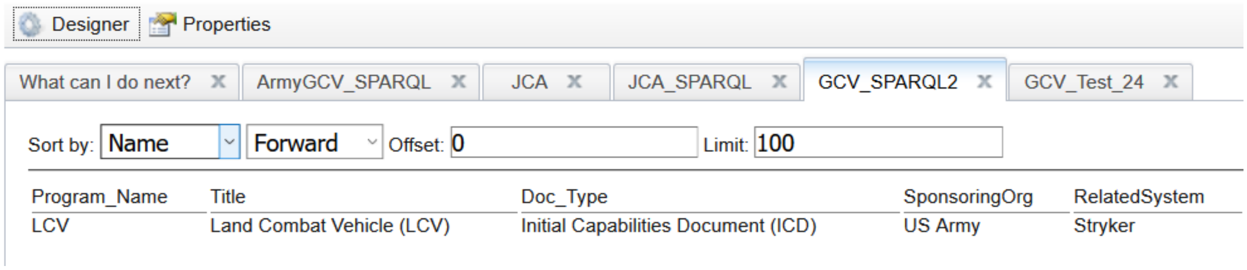


Figure 29 - Visualization of the Land Based Combat Vehicle program details as stored in the Anzo Semantic Data Lake platform

The visualization in figure 29 was created with the “Anzo on the Web” tools available with the software, and a GUI (graphical user interface) provided to build simple data representations. Figure 30 shows the “Anzo on the Web” GUI toolkit to build simple data representations from the RDF graph data inside the Anzo platform. The GUI toolkit can help any user with no knowledge of SPARQL queries to use the toolkit, and retrieve results in tabular structure from the platform. However, the limitations lie in the flexibility – it does not provide the capabilities that a SPARQL query can provide.

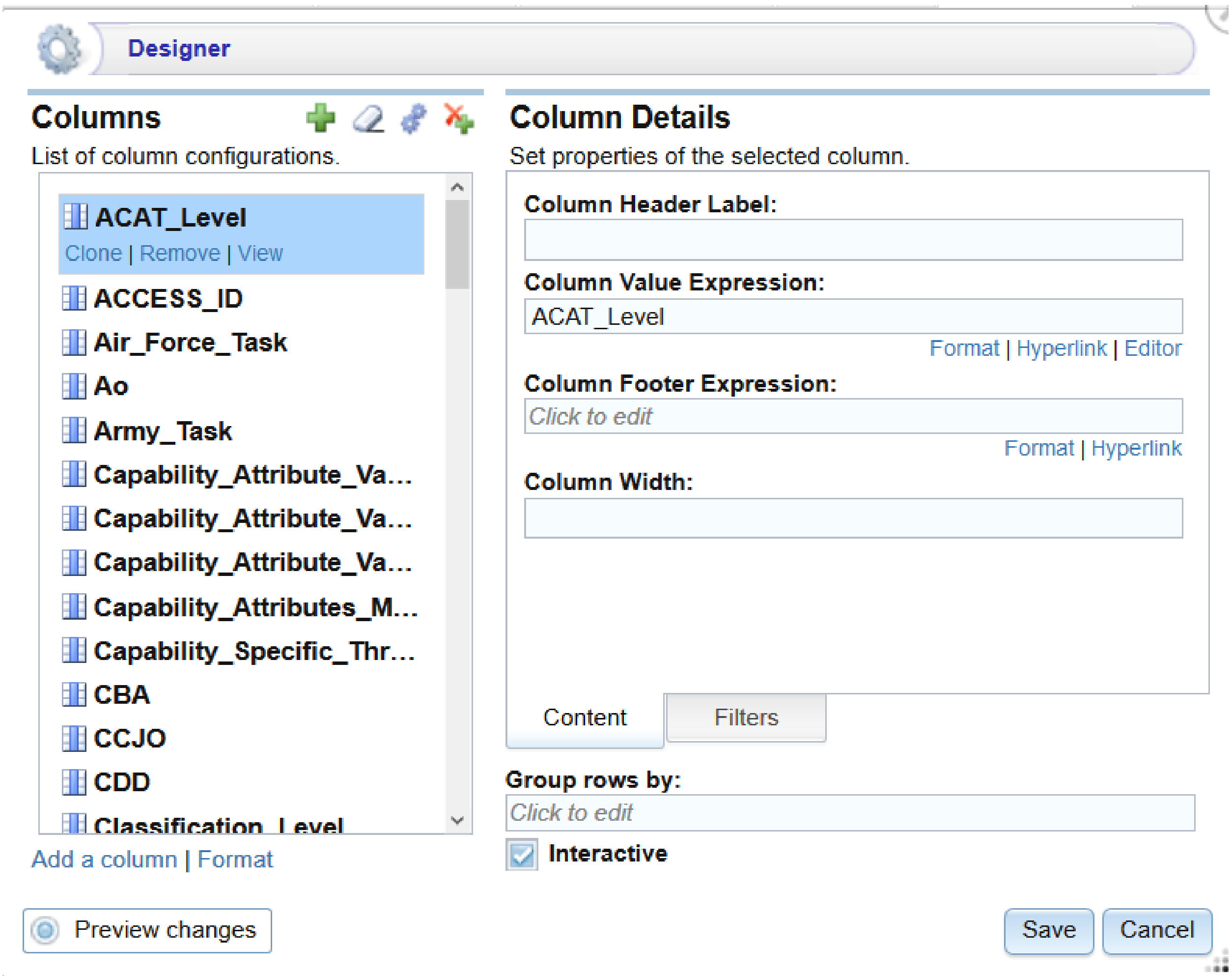
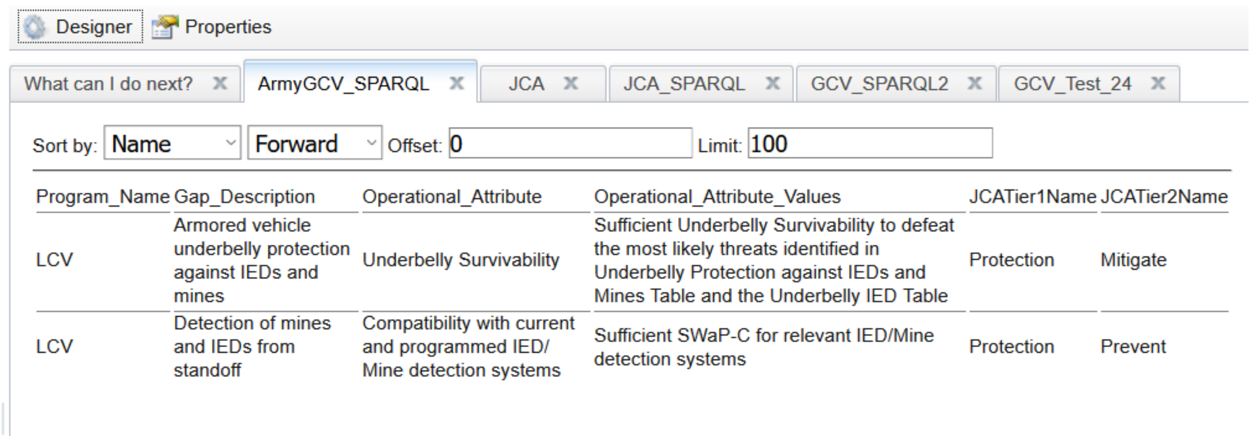


Figure 30 - "Anzo on the Web" GUI (graphical user interface) toolkit to build simple visualizations of the data from the RDF graphs in the Semantic Data Lake

As the data was fragmented when it resided in the ICD ontology documents, some of the JCIDS concepts were implicit in the documents. But, these concepts become explicit again when they are reconstructed in the semantic data lake platform. E.g., A data representation similar to a JCIDS gap table can be constructed back from the RDF triples of the data residing in the Anzo platform, as demonstrated in figure 31. Concepts like JCA tiers that apply to a certain capability lives in documents, explained in words, pictures, or tables. It is very difficult to find common capabilities based on JCA if they remain inside documents. Having these concepts in the Data Lake, creates explicit connections of JCAs with capabilities that drastically reduces the effort required to find capabilities that share same JCA. Similarly, figure 31 shows the gaps identified and the operational attributes that apply to a certain capability, retrieved from the Data Lake. To reverse track operational attributes connected with various capabilities without a Data Lake would entail sifting through large quantities of capability documents to find the mappings. Thus, when documents are broken down into data and stored in a Data Lake, the implicit concepts and connections are more easily identifiable.



Program_Name	Gap_Description	Operational_Attribute	Operational_Attribute_Values	JCATier1Name	JCATier2Name
LCV	Armored vehicle underbelly protection against IEDs and mines	Underbelly Survivability	Sufficient Underbelly Survivability to defeat the most likely threats identified in Underbelly Protection against IEDs and Mines Table and the Underbelly IED Table	Protection	Mitigate
LCV	Detection of mines and IEDs from standoff	Compatibility with current and programmed IED/ Mine detection systems	Sufficient SWaP-C for relevant IED/Mine detection systems	Protection	Prevent

Figure 31 - A JCIDS Gap Table like analogy reconstructed from the RDF triples in the Anzo Semantic Data Lake platform

4.5 Examples of insights that can be obtained from the platform

An SME analyst potentially using a semantic data lake platform to analyze JCIDS - JCEA documents and data can perform targeted queries to obtain key insights about the system of systems environment, which can be used in critical decision-making scenarios like – defining key interdependencies when proposing new capability requirements, and whether the integrating requirements are taken into account. E.g., When defining capability requirement for the Joint Future Theater Lift (JFTL) capability, the dependency on a common airframe replacement from the C-130/C-17 system, the ability to conduct Mounted-Vertical-Maneuver (MVM), and the employment of AGM-114 HELLFIRE air to surface missile. Some of the examples discussed in the following sub sections demonstrate the ability to use SPARQL query to slice, dice, analyze and present the data in a coherent format. [16]

4.5.1 Legacy Program Information Encoding and Continuity

Looking back at the JCM/JAGM program, we see that the JCIDS architecture didn't even exist when the JCM program was launched in 1999. The JCM was intended to be an all new air to surface weapon for use by the joint and allied service manned and unmanned aircraft to neutralize high value stationary, moving, and relocatable land and naval targets. But the JCM program was cancelled and reinstated several times, transforming the JCM program to the now known JAGM program. JCM would provide a common, multi-mode weapon capable of satisfying the needs across the joint platforms. The Army's inventory of TOW 2A, TOW 2B and HELLFIRE missiles was intended to be replaced by 73,000 Common Missiles, with low-rate initial production scheduled for FY08 and first unit equipped in FY10. [5] JCM's technological advances would include a tri-mode seeker: semi-active laser, millimeter wave radar and focal plane array radar. [5] The JCM's physical configuration was expected to be six inches in diameter, 50 inches long and weigh no more than 70 pounds. [5]

When the JAGM program was revised to its most recent version, capabilities were redefined. Not all original capabilities were retained in the revised plan. The capability development plan was broken down into 3 phases, as discussed in details in section 1.1. With so many continuous changes happening on a single program, there was lot of information generated in the process. E.g., Seeker types, warhead types, change in Key Performance Parameters, etc. Assuming that these changing information are captured in easily searchable, retrievable, and connectable format, features such as search, retrieve, and connect would be available off the platform. E.g., if a Semantic Data Lake has a taxonomy of seeker types, a taxonomy of warhead types, then the data can later be updated with more information such as - mapping with a new taxonomy (JCA). A snippet of one of the JCA tier 1 upto tier 4 that would apply to the JAGM program is shown in figure 32.

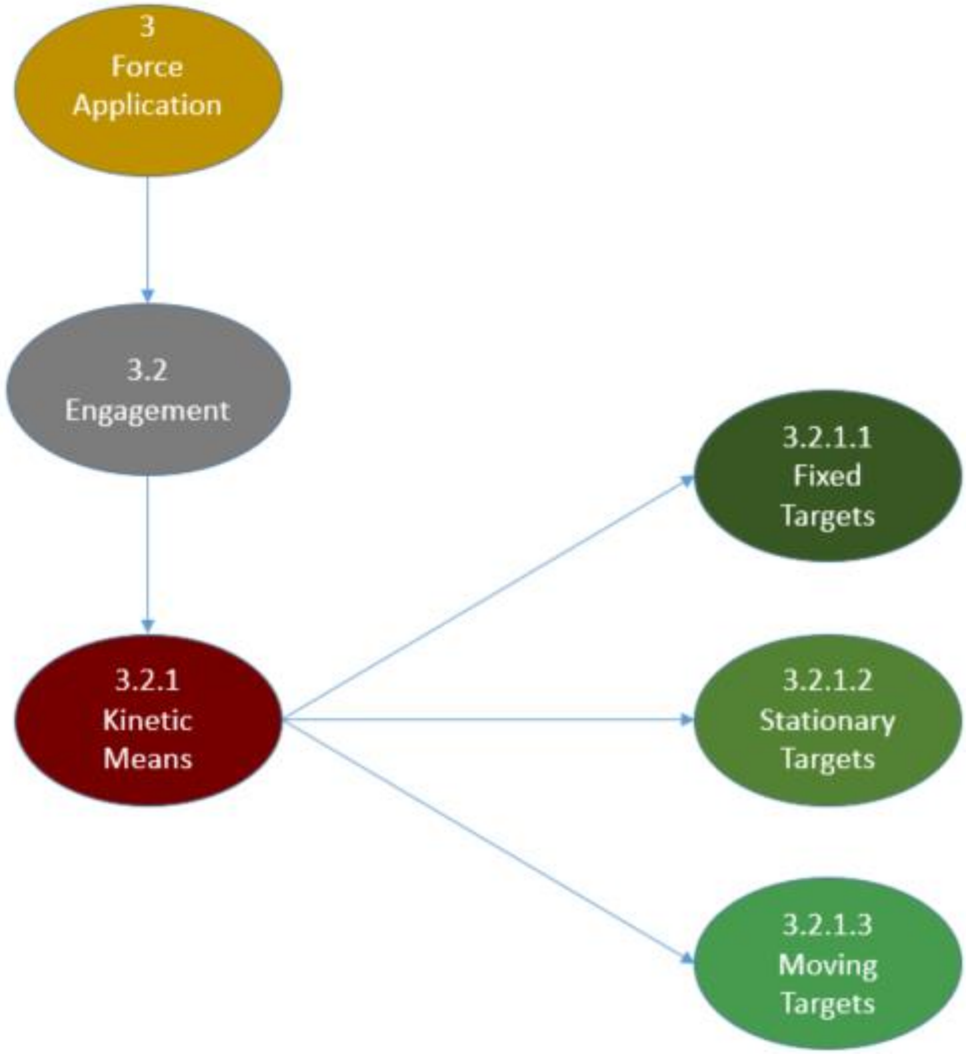


Figure 32 - JCA Tier 1, Tier 2, Tier 3, and Tier 4 breakdown for JAGM program

Figure 32, shows the JCA category scheme for targets – what type of targets can the JAGM system handle. Now, while the detailed mapping information is not illustrated here, e.g. what seekers and what warhead maps to which JCA, but it can be understood that additional general information can be added to the information residing in a semantic data lake. Such mapping would enhance the already existing information about a system in the data lake. An illustration of JCA mapping to the JAGM system with the "predicate" explicitly marked, is shown in figure 33.

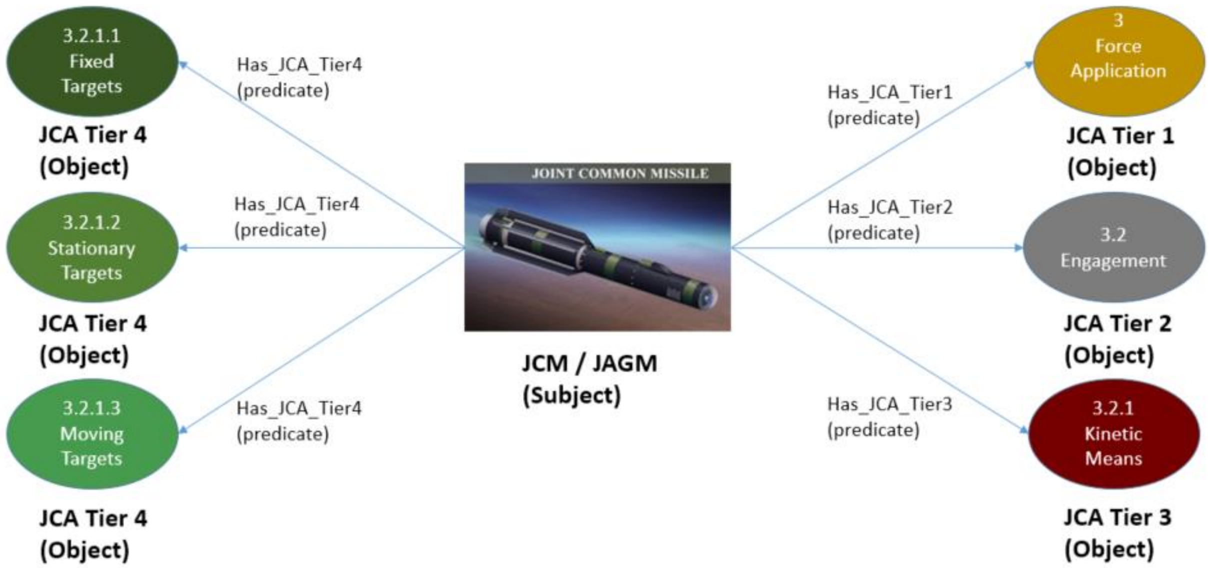


Figure 33 - Addition of JCA information to an already existing program information in the Data Lake

Assuming that the JAGM program information already exists in the Semantic Data Lake, additional information can be added to the Data Lake at a later point in time. This perfectly suits the JCM-JAGM case, where the program was started much before JCIDS architecture was constructed, and so if details of such a program existed in a Semantic Data Lake, it could be enhanced by joining additional data concepts like JCA. This can be further extended to updating the JCA ontology in the Data Lake when the next revision of JCA list is released by the DoD. Such flexibility can be provided by a Semantic Data Lake, as against a relational system, which needs major changes in the database structure to update existing schema.

Figure 33 shows a high-level view where the JAGM program is the "subject", and the tiers of JCA would be the "objects", and the subjects share a predicate relationship with the objects. E.g., JAGM (Subject) -> Has_JCA_Tier1 (Predicate) -> 3 Force Application (Object). In a semantic data lake used for such a case, the JAGM system would be a higher-level base class, and the features of the system, e.g., guidance and seeker mode, warhead type, etc. would be the properties under the JAGM system. Further granular features such as seeker type for JAGM originally thought of were – millimeter wave radar, focal plane array radar, and semi-active laser would be children under the seeker feature of JAGM. Now, each of these seeker types can be mapped to a very specific JCA, e.g. JCA tier 6 or tier 7, thus pouring additional information to already existing data in the Data Lake. In this way, more and more information can be added or deleted to model the exact real life changes that happen with a program or the surrounding structures – JCIDS, DoDAF, etc.

Now, given that other systems' information also exists in the Data Lake, this information can now be utilized to look across systems and platforms, to do a comparative study of capabilities of various systems. E.g., existing information about the HELLFIRE missiles or the SDB (Small Diameter Bomb) II in the Semantic Data Lake can now be used to evaluate the JAGM program. The SDB II has a tri mode seeker in addition to GPS guidance. [38] These GPS and guidance features give the SDB II ability to strike, fixed, stationary, and moving targets in all-weather conditions. [38] Thus, the seeker mode features can be mapped to the JCAs as shown in figure 33, JCA tier 4 – 3.2.1.1 (Fixed Targets), JCA tier 4 – 3.2.1.2 (Stationary Targets), and JCA tier 4 – 3.2.1.3 (Moving Targets). Therefore, by comparing the JCAs of JAGM and the SDB II program, it can be inferred by an analyst that there is overlap of capabilities in the JAGM and the SDB II programs. This provides a decision support ability, for the SME to evaluate various systems across services, and find synergies to maximize avenues for value from investments.

This is just one dimension of capability requirements analysis. There can be multiple other context of comparisons, i.e. the properties based on which two or more systems need to be compared, such as weight, size, warhead etc. Each dimension can provide unique ability to compare and contrast capabilities. This is what a Semantic Data Lake can provide as an intrinsic ability, rather than make stop gap arrangements with a relational solution. SME analysts can compare systems and programs using lens such as JCA. Once JCAs are associated with various systems, business rules can be written on top of the Data Lake to create higher-level inferences. E.g., a hypothetical business rule that assumes that if a weapon system can successfully strike a moving target, then it also can successfully strike stationary and fixed targets. This is illustrated in figure 34.

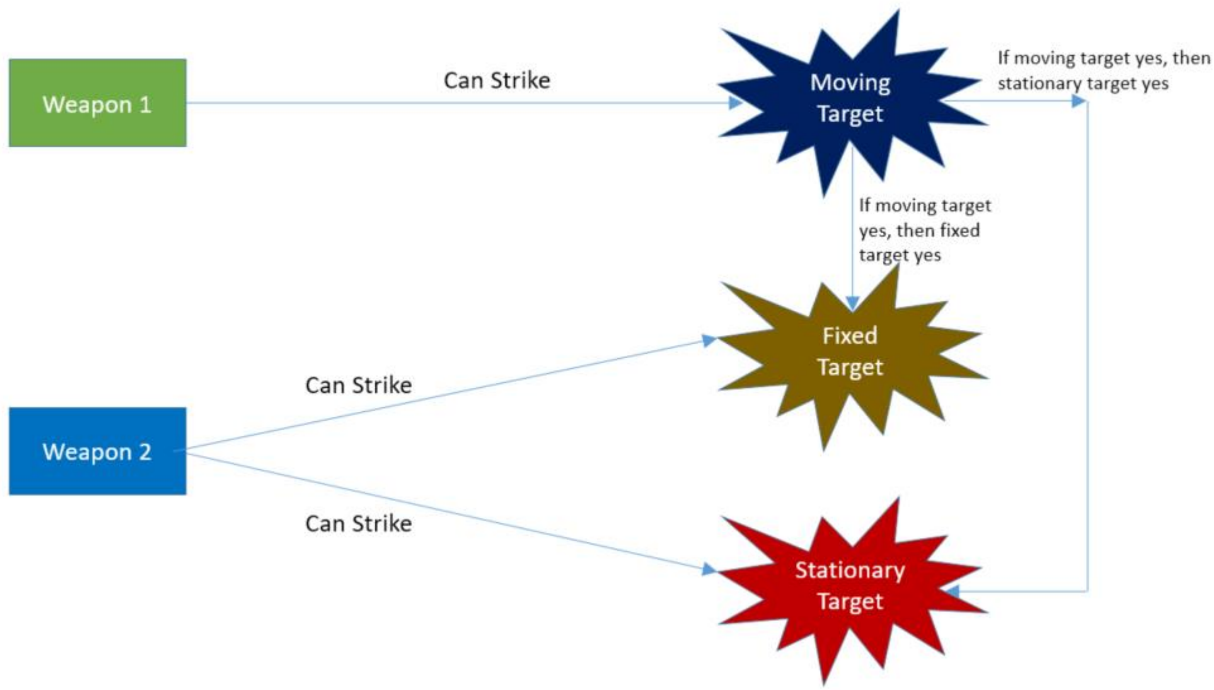


Figure 34 - Illustration of a transitive relationship

Figure 34 shows an example of a transitive relationship, where weapon 1 can strike a moving target and a business rule exists that if a moving target can be hit, then a stationary as well as a fixed target can be hit. This is called a transitive relationship, where if a binary relation x to y is true, and a binary relation y to z is true, then under a transitivity, binary relation x to z should also be true. It can be written in the formal notation as shown below.

$$(\forall x)(\forall y)(\forall z)((\boldsymbol{R}(x,y) \ \& \ \boldsymbol{R}(y,z)) \rightarrow \boldsymbol{R}(x,z))$$

Applying transitive property to the relationship described in figure 34, it can be inferred that weapon 1 can also successfully strike fixed and stationary targets. But the converse may not be true, i.e., weapon 2 that can successfully strike fixed and stationary targets, may not necessarily be able to strike moving targets. The transitivity relation is a simple example of how business rules can be codified into logic, hooked up with the data in the Semantic Data Lake, and additional insights can be gained using mathematical inferencing. Such capability cannot be easily obtained with relational database systems.

The main value derived out of a Semantic Data Lake in cases such as the JCM/JAGM program is the information continuity. For long running programs, various aspects change, the contextual information changes, requirements change and evolve, personnel change, but this gradual change or evolution is not captured with information. Therefore, a flexible, granular data based Semantic Data Lake can capture various information at different time points, and provide a unifying view to a new SME analyst. This can help the SME analyst concentrate on the actual job at hand – evaluate programs and capability requirements, rather than data collections and sifting through data to find relevant information.

4.5.2 Discovering Common Reference Network

Figure 35 shows an illustrative example of a common reference used in the ICD documents for two separate systems – Joint Future Theater Lift (JFTL), [39] and Joint Direct Support of Aerial Intelligence Surveillance and Reconnaissance (JDSAISR). [40] This illustrates that a reference network can be built similar to a citation network. A reference network can help find common lineage, or common areas of usage. If more than one ICD document refers the same source, then there is a degree of commonality in some sense between the systems or programs. Such networks can be built and stored without much meta information using semantic data lake solutions. Compared to a relational database equivalent to express the same relationship would entail frequent change in the schema, and still, it may not entirely capture the "semantics" of the relationships.

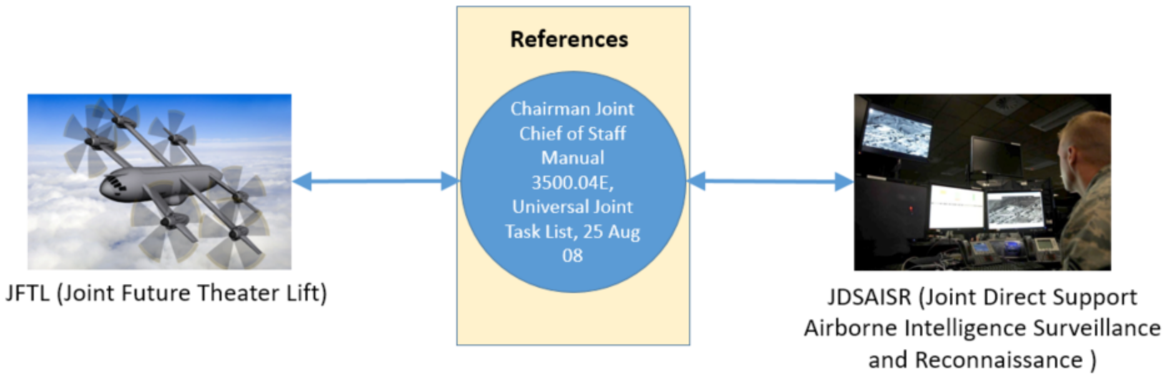


Figure 35 - Illustrative example of common references in ICD of the JFTL (Joint Future Theater Lift) and JDSAISR (Joint Direct Support Airborne Intelligence Surveillance and Reconnaissance) programs

Image Credits - [41] [42]

This data was discovered using a SPARQL query feature. The SPARQL query used to develop the reference comparison between the JFTL and the JDSAISR systems is shown in figure 36. The query looks for the ICD ontology concept – reference, and pulls out the names of the programs that refer to the particular reference. (1501 is the ontology slot tag of the ontology concept Reference). In this example in figure 35, the two systems are shown to be sharing the JCIDS manual. This is a trivial example, but this example is used as a proof and illustration that common references between different systems can be mapped and compared.

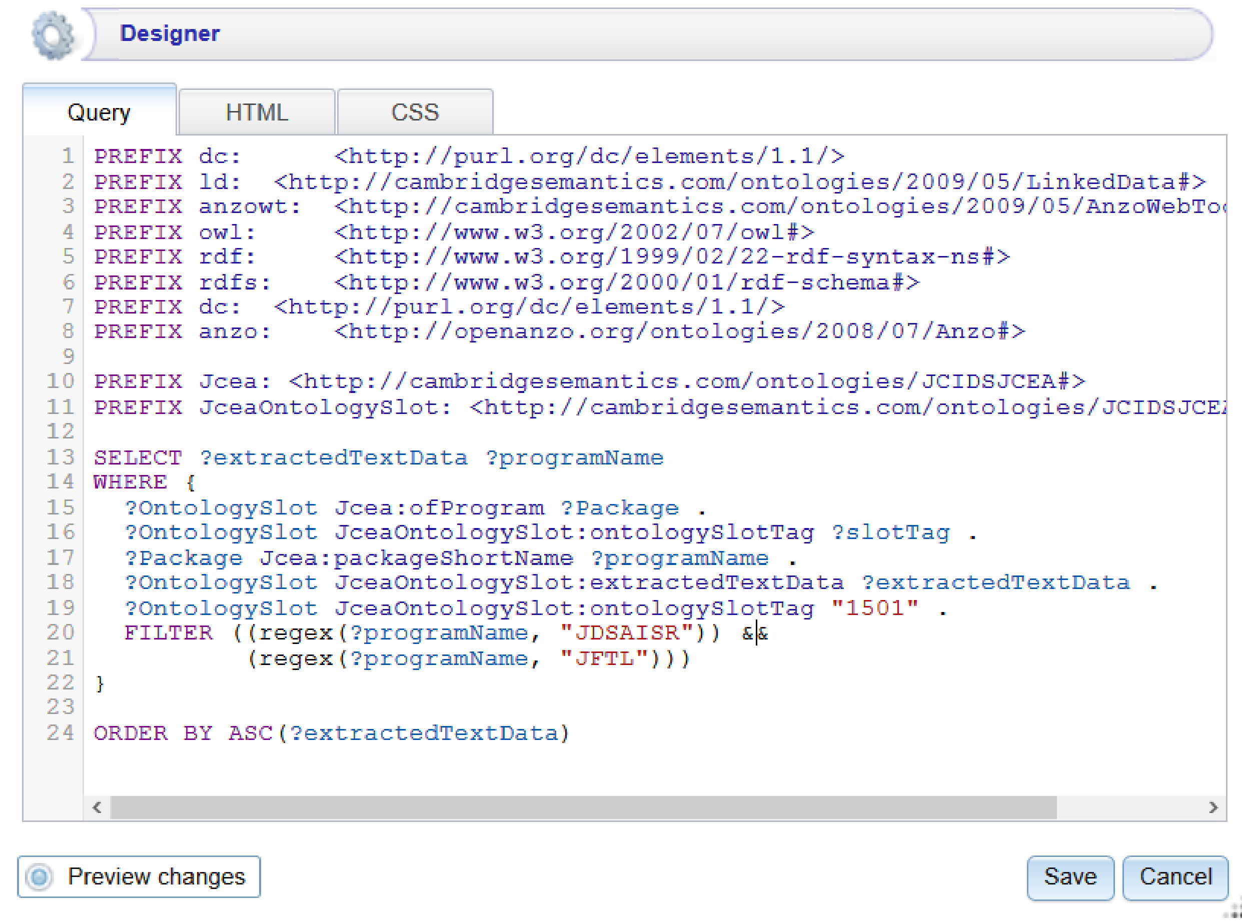


Figure 36 - SPARQL Query used to discover the reference network for JFTL and the JDSAISR programs

The SPARQL analysis revealed a commonality of 8% in the references of the ICD ontology for the programs JFTL and the JDSAISR. Similar quantitative analysis can be performed on other systems against other dimensions (e.g., operational attributes, gaps, etc. instead of references). Such detailed quantitative analyses are not well suited with a database system like the relational database system.

4.5.3 Identifying Systems with Common Capability Areas

Figure 37 shows an example of a subset of systems that share the Joint Capability Area tier 1 – battlespace awareness. The F22 Raptor (Fighter Aircraft), F15 Eagle (Fighter Aircraft), Stryker (Armored Personnel Carrier), NAVSTAR (Global Positioning System), Sentinel Radar (Short Range Air Defense Radar), and THAAD (Terminal High Altitude Air Defense system) – disparate systems, but share a common capability area – battlespace awareness. This is critical in sharing information between fielded systems, because weapon systems need to collect information about the target, or hostile incoming systems prior to engaging them. This can be extended all the way down to JCA tier 7, and a map of common systems (similar to the one depicted in figure 38) can be created. This can help a SME analyst easily identify existing systems that share a capability area when defining an ICD or analyzing competing solutions. The classic JCM-JAGM case discussed in section 1.1 can be applied here. An analyst working on a resurrected version of the JAGM program, can compare existing capability solutions that has the potential to achieve similar effects. HELLFIRE missiles and the SDB-2 glide bombs can emerge as one of the contenders with an analysis on the common JCAs. This can be taken a step further by comparing the KPP of the existing solutions and the proposed capability requirement to determine how much of a performance benefit the new capability can provide, and ultimately, whether it is worth the effort to pursue research and development efforts and money on such a program.

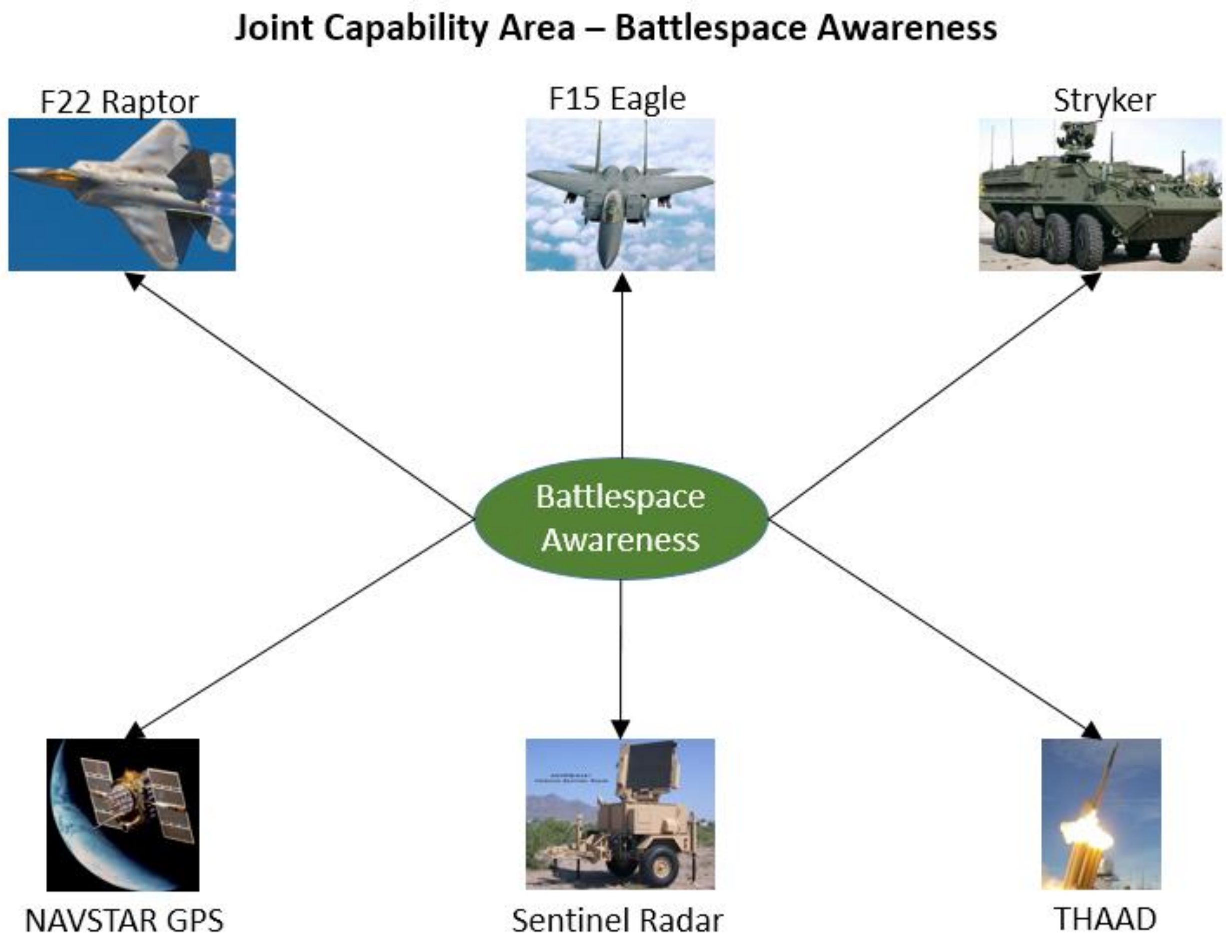


Figure 37 - Some systems that share the JCA Tier 1 - Battlespace Awareness

Image Credits - [43] [44] [45] [46] [47] [48]

Snippets of JCA tiers 1 to 9 are shown in Appendix D. There are many lower level JCAs under Battlespace Awareness that can be mined to find commonality across systems as shown in figure 37. Capability areas such as imagery collection, and the technique used in imagery collection can reveal granular capability dependencies and capability overlaps of different systems.

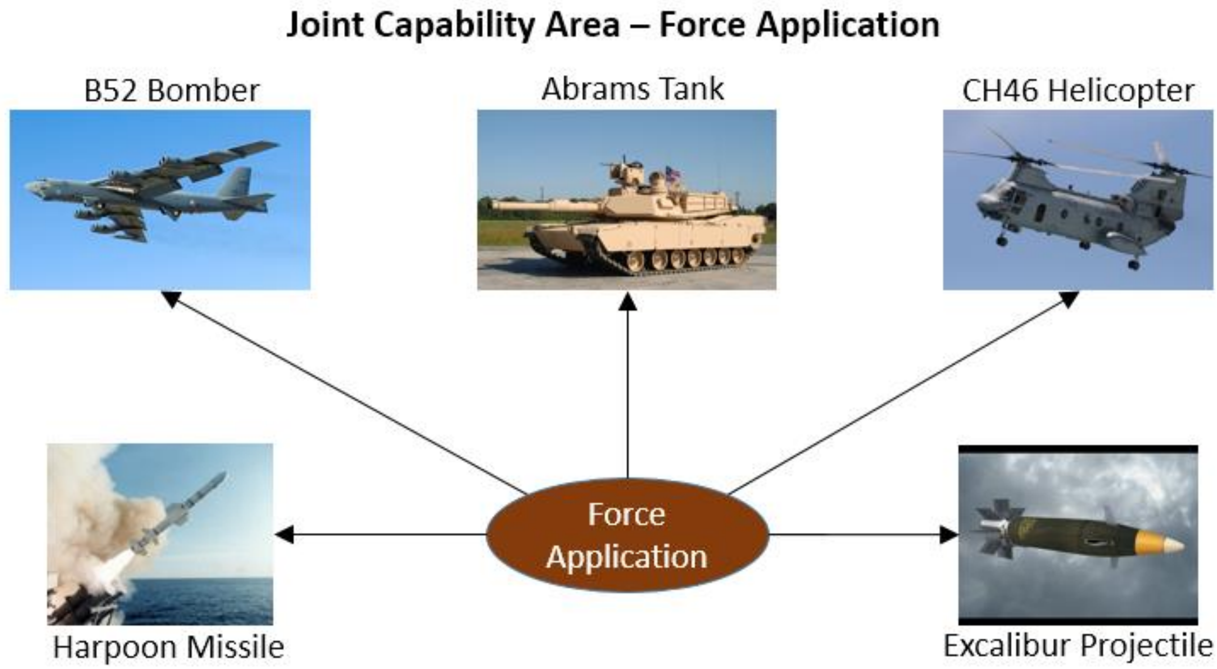


Figure 38 - Some systems that share the JCA Tier 1 - Force Application

Figure 38 depicts illustrative results of the analysis of querying systems that share the JCA tier 1 – Force Application. A subset of the retrieved results is shown in figure 38. The systems shown are – B52 (High altitude bomber), Abrams Tank, CH46 (Heavy Lift Helicopter), Harpoon Missile (Sea Launched Attack Missile), and the Excalibur Projectile (guided artillery shell). Although the results shown here are highly varied, but it serves as an illustrative example of the range of alternatives available to fulfill a certain capability area – in this case, force application. These analyses can be performed upto the deepest layers of the JCA, and help the analyst make informed decision on capability requirements, taking into account the inter systems dependencies, analysis of alternatives, etc.

In addition, the semantic data lake is not fixed to a certain data structure ontology defined in the system, and can be updated to suit the dynamic needs of the environment. For example, during the 15 years from the original JCM to the present, the categories used to classify capabilities have changed several times. At the time the JCM ICD was written, JCAs had not been developed. The ICD uses analogous concepts such as Joint Force Application. Since then, JCAs have been revised several times. These changes to the JCAs and other concepts can be updated in the semantic data lake to reflect the changes, without making any changes to the existing data in the systems. (Updating schema in relational databases can be quite a task).

4.5.4 Identifying Programs, Capabilities or Systems Fulfilling Common Mission Areas

One of the important questions that a decision support system used with JCIDS-JCEA should be able to answer is how do these systems work together to accomplish a given mission? [3] The key to linking systems to missions are the Universal Joint Tasks (UJT). [3] The UJTs provide a list of tasks, or warfighting functions, that a program or organization executes to support the joint warfighting mission. [3] The UJTL supports DoD to conduct joint force development, readiness reporting, experimentation, joint training and education, and lessons learned.

UJT_Id	UJT_Name	System_Context
OP 1.2.4	Conduct Operations in Depth	AH-64E Apache New Build
OP 1.2.4	Conduct Operations in Depth	AIM-120 Advanced Medium Range Air-to-Air Missile
OP 1.2.4	Conduct Operations in Depth	Hellfire Air to Ground Missile
OP 3.2.5.1	Conduct Air Interdiction of Operational Forces/Targets	USA Infantry Brigade Combat Team
OP 3.2.5.1	Conduct Air Interdiction of Operational Forces/Targets	USA Air Assault Brigade Combat Team
OP 3.9	Conduct Target Validation	USN Cruiser Organization
OP 3.9	Conduct Target Validation	USN Destroyer Squadron
OP 3.9	Conduct Target Validation	USN Helicopter Sea Combat Squadron

Table 7 – Example of UJT shared between a subset of various types of systems

Table 7 shows an example of a subset of systems sharing the same UJT. The different capabilities or systems operating on common mission areas have the potential need to synchronize and the need of interoperability. E.g., in table 7 under UJT OP 1.2.4, the system AH-64E, Apache attack helicopter shares the same UJT OP 1.2.4 – Conduct Operations in Depth with the Hellfire missile system. This is an illustrative simple case depicted here. Not all results will be related, but it can greatly help the SME analyst looking for commonality in mission areas across systems, programs, and capabilities.

Such varied comparisons can be performed over a variety of breadth of systems to unleash the collective, macro, and a sociotechnical viewpoint of what is the current state in the whole DoD from a capabilities point of view, what is desired, and what are the gaps, expressed in terminologies such as JCA, UJTL, etc.

4.5.5 System or Program Interdependencies

The fact files of the US Air Force and the Navy were another important piece of the experiments performed. These fact sheets contained some key characteristics and performance information about the systems in the inventory. Trying to find dependencies by intelligently firing targeted SPARQL queries across data about the various platforms can reveal some interesting dynamics about cross platform integration. The dependency can be defined as a constraint that needs to fulfill in order for two systems or platforms to interoperate with each other.

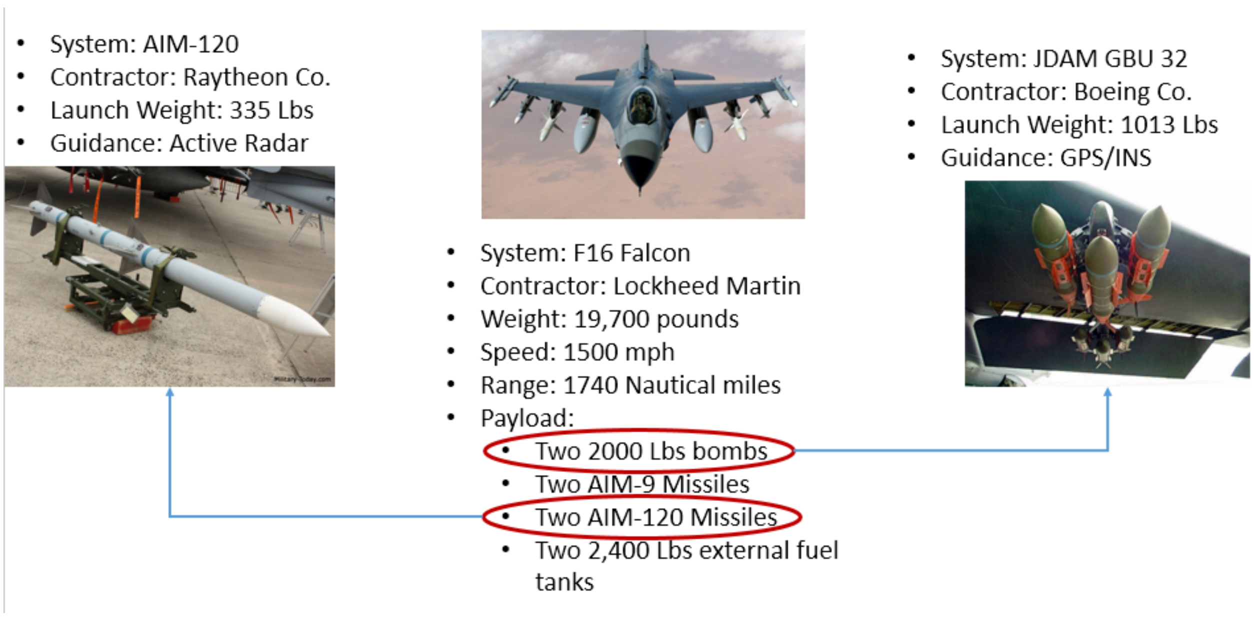


Figure 39 - Data from the Air Force fact files shows interdependency on systems type (AIM-120) and weight category for interoperability

Image credits - [48] [49]

Facts - [50] [51] [52]

Figure 39 shows how the key platform information about a system has direct implication about the dependencies on other systems. F16 Falcon fighter has a payload capacity of two AIM 120 missiles and two 2000-pound bombs. Querying other system data can uncover that the AIM 120 missile is directly compatible on the F16 platform, and so is the JDAM GBU 32 bomb. This has been achieved with a targeted SPARQL query for the PoC. The logic used to find the interconnection between the F-16 and the JDAM GBU is shown in figure 40. The F-16 has a systems property of payload 1, which is of type bomb. The maximum weight of the payload 1 is defined by the attribute weight. The JDAM system is of type bomb, and has a property weight.

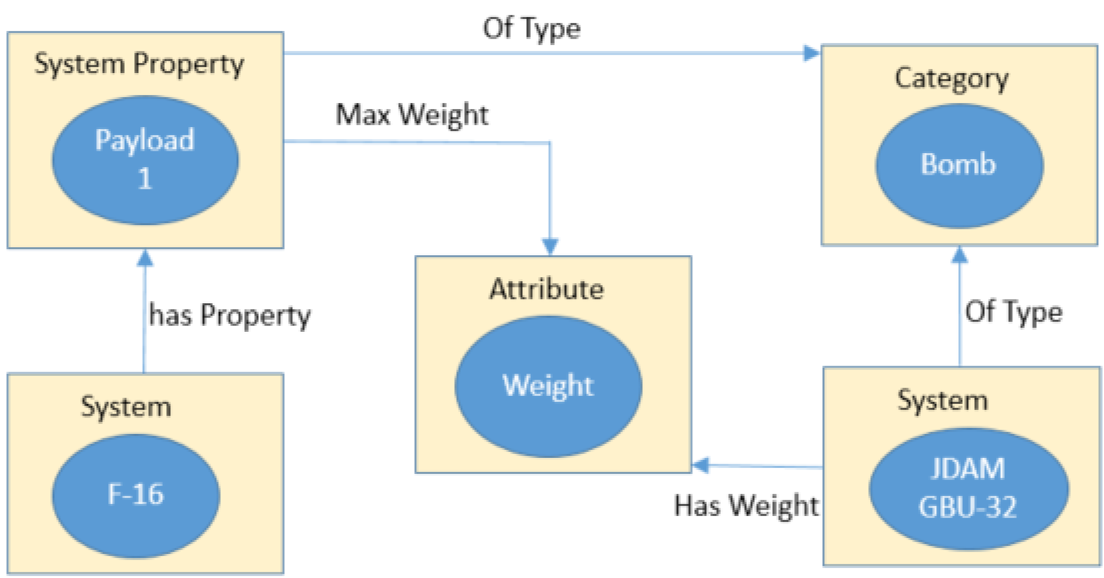


Figure 40 - Semantic relationship diagram of the dependencies of F-16 and JDAM GBU as described in figure 39

What was achieved in this example was performed with a targeted SPARQL query, knowing beforehand what was to be looked for. But the idea is that, the same logic can be codified with scripts, and even further automated with reasoning/inference engines for easy discovery of dependencies.

5. CONCLUSION & RECOMMENDATIONS

This thesis presented proof of concept (PoC) experiments performed to evaluate the application of Semantic Data Lake technology to the problem of creating basic elements of a JCIDS Joint Capabilities Enterprise Architecture (JCEA) for capability requirements portfolio planning in the United States Department of Defense.

The first experiment was transforming a capabilities requirements ontology from a linear Excel spreadsheet form into an equivalent OWL representation, which was then loaded into the Anzo Semantic Data Lake technology. The next experiment was using the ontology to transform and load spreadsheet data previously extracted by subject matter experts (SMEs) from capability requirements documents. The first two experiments demonstrated that essential elements of sample capabilities requirements documents could be organized against an ontology and stored in a Semantic Data Lake system. A third experiment loaded data from a relational database containing data for a subset of the concepts in the ontology that had been extracted from a larger sample of capabilities documents.

In order to demonstrate integration of separately sourced data, the next set of experiments showed that common taxonomies and other frameworks could also be loaded and integrated into the Semantic Data Lake. These included the Joint Capability Area (JCA) taxonomy that is used to classify and compare capabilities and the Universal Joint Task List (UJTL) that is used to specify what a capability does.

Having shown that representative JCEA data can be loaded into a Semantic Data Lake, the next set of experiments looked at how data can be presented to users. Data is stored in the Semantic Data Lake in RDF using fine-grained subject-predicate-value triples. The SPARQL query language extends concepts from the relational SQL language for use against an RDF data store. The Anzo software used in the experiments came with a web tool called "Anzo on the Web" which automatically generates SPARQL queries against the data stored in RDF graph format for presentation in two-dimensional tabular format. Reading and understanding tabular data with subject matter knowledge doesn't require special skills, and so this illustrates that a user without any deep understanding of semantic web technology can comfortably use the technology with robust software like Anzo.

The next step explored the analysis of the stored data in the Semantic Data Lake that can aid in augmented decision-making. Various JCIDS-JCEA architecture concepts were validated using semantic SPARQL queries. JCIDS concepts like JCA, UJTL, and connected references to other documents, were used as a pivot to compare and contrast disparate capabilities. These experiments demonstrated the potential for the a JCEA Semantic Data Lake to assist SMEs in more rapidly discovering knowledge, such as inter-systems dependencies, that would otherwise haven take a lot of manual sifting through thousands of documents. This shows that the JCEA Semantic Data Lake can assist in narrowing the scope of search and discovery.

These experiments validated and verified the hypothesis that the JCIDS JCEA architecture can be supported on a Semantic Data Lake platform, that the information extracted by SME analysts and that additional information added from other sources to the Data Lake can enhance the knowledge and fact representation on the platform. Information from disparate sources can be added to the Data Lake repository without many inherent changes, such as updating the ontology. These experiments also showed that if JCIDS, JCEA, and other DoD architectural data about the retired, fielded, and in-development programs, are aggregated into the Semantic Data Lake, there can be numerous focal points and lenses to evaluate different capabilities. Thus, such a Semantic Data Lake platform can aid an SME analyst, or other end users of the platform, with critical information, which is easy to assemble, and use it for capability requirements, portfolio management analysis.

5.1 Limitations and Challenges

One of the challenges that were discovered when conducting the PoC experiments was that it requires considerable knowledge of the semantic technology to setup a platform that can be used by the DoD users efficiently. It may require considerable technology investment to install, configure, and acquire user licenses for software (e.g., Anzo) that supports the Semantic Data Lake platform. It was also noted while using Anzo software that the built in tools are easy to use for the non-tech users, but also had limitations in its capabilities. To utilize more powerful features such as SPARQL queries, the user needs knowledge of the underlying ontology and RDF representation. A major observation is the nature of ontology that is built for supporting the US DoD capability requirements management. A more structured and descriptive ontology can help in capturing more structured data, which leaves the platform less flexible to quick changes in the type of data being fed into it. E.g., a robust JCA ontology would need more work to change if there is a revision to the hierarchy of the current JCA taxonomy, whereas a loosely defined ontology can provide more flexibility to support future changes to the ontology, but would leave the logic expression of the ontology less powerful.

5.2 Future Research

This research effort and the thesis has developed proof of concepts, or simple prototypes in the realm of the limited data available to demonstrate how a semantic data lake platform can be setup for supporting the capability requirements portfolio management problem with decision support system ability. As such, the research was unidirectional and limited. Although the results validated the premise, the resulting prototype and Semantic Data Lake is rudimentary with limited value outside this academic effort. However, this thesis shows the potential of results that can be achieved with such a system. More work is required to quantify the benefits that can be gained from an automated Semantic Data Lake platform with Decision Support System abilities integrated with an inferencing engine. The platform can be integrated with an inferencing engine or a reasoner such as Apache Drools, [54] Apache Jena Reasoner, [55] Pellet. [56]

The examples covered in figure 39 and figure 40 can be expanded to include semantic reasoning. This PoC demonstrated the potential that underlying logic of interconnecting systems can be encoded as targeted SPARQK queries on the Semantic Data Lake. Discovering compatibility of the GB-32 and the AIM-120 based on the weapon system properties compared to the delivery platform, F-16 Falcon’s properties can be done by augmenting the Semantic Data Lake with an inferencing engine. The idea is that this can be extended to look for specific properties of systems at a simple level, and use reasoning at a higher level, if desired.

Rule-based reasoning engines can also go beyond linkages based on data connections (such as JCA and UJTL) to pinpoint areas of improvements across the capabilities requirements portfolio and areas for cooperation across the different services and programs. E.g., Areas of synergies can help reduce complexity, and use common platforms for development, like in the case of the JAGM program, HELLFIRE missile upgrades, and the SDB II. Reasoning about connections among capability requirements can also help in discovering alternatives when unforeseen events occur, such as budget or performance driven program cancelation or changes in threats. The ability to identify systems with similar capabilities will help enable synergy between different programs, identify redundant capability overlaps, and streamline investments to focus on programs that have the greatest exclusivity from each other to maximize the value benefit.

APPENDIX A – Joint Air-Ground Missile (JAGM) Key Events

Key events of the revised JAGM program in chronological order is presented below.

Year	JAGM Program Key Events Overview
1999	JCM (Joint Common Missile) program inititated
2000 - 2003	Raytheon, Tucson, Arizona, Lockheed Martin Missiles and Fire Control, Orlando, Florida, and the Boeing Company, Huntsville, Texas, began risk reduction, concept and technology development work which was completed in FY03
2003	JCM ICD released
2004	The program was in the Technology Development (TD) phase with several competing contractor teams preparing for Milestone B. Lockheed Martin won an order to develop and build the Joint Common Missile for the Army, Navy and Marine Corps, beating Boeing and Raytheon in the competition. In December 23, 2004 the pentagon announced plans for saving of $30 billion between 2005 and 2011. Among the programs cancelled due to these cuts is the $2.4 billion funding for the JCM program
2007	A new JAGM (Joint Air-to-Ground Missile) program is launched as a successor to the Hellfire missile and the terminated Joint Common Missile Program
2008	27 Month, $125 million technology development contract for the JAGM program announced. The contract funds a program to develop and fire 3 prototype missiles with fully integrated tri-mode seekers.
2009	Boeing subsidiary McDonnell Douglas Corp. in St. Louis, MO received a $7.4 million time and material delivery order against a previously issued Basic Ordering Agreement (N00019-05-G-0026) for wind tunnel testing of JAGM prototypes on their F/A-18E/F Super Hornet. Raytheon Co. in Tucson, AZ receives an $18.7 million fixed price incentive firm target contract, for 27 months of technology development for the Joint Air Ground Missile Program. Lockheed Martin Corp. in Orlando, FL received an $18.7 million fixed price incentive firm target contract, for 27 months of technology development for the Joint Air Ground Missile Program.
2010	Raytheon, Boeing and Lockheed Martin complete a series of tests from various platforms like the F/A-18 E/F Super Hornet and a ground-based rotary-wing launcher to validate various parameters of the missile.
2011	DoD Buzz reports external link that Raytheon isn’t using a production version of the JAGM missile in its firing tests, just the seeker. Raytheon replies that the tests’ terms are aimed at the seeker, and do not require production-ready missiles. Lockheed Martin says that their JAGM test missiles have all been production ready configurations – but that will only help them in the short term if failings in their test firings are traceable to their missile design, rather than their seekers.

2012	A March 2012 presolicitation external link from the US Navy for JAGM integration on F/A-18E/F aircraft may have sent mixed signals, but its cancellation confirms the Navy's intent. In its 2012 Selected Weapons Program assessment report external link, the GAO (US Government Accountability Office) underlines the uncertain nature of JAGM's future – not quite cancelled but close. It notes that Hellfires have been working well in theater, weakening the case for an expensive replacement. Frank Kendall, undersecretary for acquisition, technology, and logistics, signs an Acquisition Decision Memorandum, granting new life to the JAGM program. Meanwhile, the Army has produced a JAGM affordability study, and provided it to the 2 teams. In the FY2013 Presidential Request, the US Navy estimates it is a "manageable risk to terminate the Navy's and USMC's investment in the JAGM program," choosing to invest instead in SDB II and continued Hellfire procurement.
2013 - 2015	Lockheed Martin has been awarded a $66.4 million contract external link to further develop the Joint Air-to-Ground Missile (JAGM) for the Army and Navy. The USMC's Aviation Plan to 2030 deals with weapons as well. JAGM is mentioned, and its 3 planned increments are fully outlined. Under current plans, JAGM Increment 1 will begin integration with AH-1Z attack helicopters in 2015, and will achieve Initial Operational Capability on the AH-1Z and on KC-130J Harvest Hawk armed tanker/ transport planes in 2019. Beyond 2019, the USMC plans to field JAGM Increment 3 on the AV-8B Harrier II and F-35C Lightning II. The US military slowly files its budget documents, detailing planned spending from FY 2014 – 2019. According to those documents, AGM-114 Hellfire orders stop in FY 2015 (USAF), and the last Hellfires will be delivered in April 2017. The Army's documentation says nothing about JAGM production, except that the Milestone C decision for low-rate production is expected in Q2 FY17. Meanwhile, Navy documents indicate that they're back in the program. They show JAGM integration on AH-1Z helicopters beginning in FY15, and orders beginning in FY19. JAGM will be re-using most of the AGM-114R Hellfire, which is already integrated on the AH-1Z, but Navy helicopters are used to the video interface that JAGM won't have, and don't typically carry fire-control radars.
2016	The US Army has successfully fired external link Lockheed Martin's multi-mode Joint Air-to-Ground Missile (JAGM) from a UAV for the first time. Testing was conducted on an MQ-1C Gray Eagle at Dugway Proving Ground, Utah. The Gray Eagle test was the seventh flight test for the JAGM missile. The missile was previously tested on Apache attack helicopters and Marine Corps Cobra helicopters.

APPENDIX B – Detailed Discussion of Gaps under JCIDS Process

Some of the gaps under the JCIDS process are –

1. Affordability

Affordability is a key challenge facing JCIDS. Since 1995, the Army alone has cancelled "22 major programs, at an estimated cost of $32 billion, for equipment that was never built or fielded". [57] Upfront, JCIDS has to determine affordability. [16]

The goal of the updated JCIDS process is to achieve better affordability through: [16]

- Up front analytical rigor
- Greater fidelity in cost, schedule, and estimates enabling better performance trade-offs
- Incorporation of affordability as a firm requirement in all programs like other "ilities"
- Making the tough decisions up front and throughout the lifecycle

2. Requirements Instability

Requirement instability results in costly schedule delays and performance issues later in a program's life. Starting with the Warfighter's capability analysis, the hard work must be done up front and remain stable throughout a program. To improve stability, JCIDS must ensure requirements are captured up front in the process and eliminate requirement creep by establishing mechanisms such as configuration steering boards to limit future changes. [16] Requirements form the basis of any program and JCIDS is the mechanism by which those requirements achieve stability. Without stable requirements, programs will never achieve their intended purpose and this is not an option in today's environment. Stability is a must. [16]

3. Tough Decision Making Early

Additional JCIDS challenges include the inability to prioritize requirements and lack of rigor required to eliminate wasteful programs. To tackle these issues, time needs to be invested up front in the JCIDS process to ensure the proper discussions occur and analysis is done. JCIDS intends to achieve these objectives through: [16]

- Limiting the decision audience (cut through bureaucracy)
- Balancing cost vs. capability vs. risk
- Making the tough decisions early
- Debating critical items such as portfolio analysis
- Reviewing the entire (all classification levels) DoD portfolio to determine if a solution exists prior to creating one
- Being solution-centric, not document/process-centric

APPENDIX C – ICD Ontology Design

The prototype ontology has fifteen clusters of concepts with data slots for collecting and organizing information extracted from capabilities documents.

1. Reference Data Cluster

The data captured in this cluster come primarily from sources outside the document itself, including the ACCESS database prepared in an earlier analysis. Data from the document may be included if present.

101	ACCESS ID (if known from prior analysis)
102	Program Name
103	JRAD Portfolio
104	PNO
105	Program Cost
106	Commodity Type
107	IOC (actual intended or assumed for planning)
108	FOC (actual intended or assumed for planning)
109	Retirement (assumed or planned)

Table 8 - Reference Data for Ontology Concept [58]

Table 8 covers the references ontology structure. These reference data are the key binding glue to the interconnectedness to other systems, programs, etc.

2. Cover Page Data Cluster

Data in this cluster usually come from the cover page of the document. When this type of data appeared in a different location in the source document, it was included here and the document section and page number it came from was identified.

201	Doc Type
202	Classification Level
203	Title (Full Program Name)
203	Sponsoring organization
205	Date submitted
206	ACAT Level
207	Proposed validation authority
208	Proposed MDA (Milestone Decision Authority)
209	Proposed JSD (Joint Staffing Designator)

Table 9 - Cover Page for Ontology Concept [58]

Table 9 covers the details found in the JCIDS documents cover page ontology structure. This has key information about the source document such as type of the document, the classification level, the service arm sponsoring the document, etc.

3. Strategic Guidance Data Cluster

This common thread integrates and synchronizes the activities of the Joint Staff, combatant commands, Services, and combat support agencies. As an overarching term, strategic direction encompasses the processes and products the President, Secretary of Defense, and Chairman of the Joint Chiefs of Staff use to provide strategic guidance in the form of various strategic products. (JP 5-0) The JCIDS Capabilities-Based Assessment (CBA) process is rooted in a chain of strategic guidance documents. The National Security Strategy (NSS), the National Defense Strategy (NDS), and the National Military Strategy (NMS) provide the overarching description of the country's defense interests, objectives, and priorities. In addition, the Guidance for the Employment of the Force (GEF), Defense Planning Guidance (DPG), and the Quadrennial Defense Review (QDR) Report contain further refinement of objectives and priorities. Together, these help provide an analytical foundation and context for the CBA as well as other capability analyses. [59]

301	NSS - National Security Strategy
302	HSHS - National Strategy for Homeland Security
303	NDS - National Defense Strategy
304	QDR - Quadrennial Defense Review
305	NMS - National Military Strategy
306	DPG - Defense Planning Guidance
307	GEF - Guidance for the Employment of the Force
308	CRA - Chairman's Risk Assessment
309	JSCP - Joint Strategic Capabilities Plan
310	CCJO - Capstone Concept for Joint Operations
311	Other strategic guidance

Table 10 - Strategic Guidance for Ontology Concept [58]

Table 10 shows the strategic guidance types. This cluster has slots for ten types of strategic guidance references. The last slot ("Other strategic guidance") has been used for any other type of strategic guidance reference.

4. Core Mission Areas Data Cluster

DOD core mission areas identified under the most recent Quadrennial Roles and Missions (QRM) review are: Homeland Defense and Civil Support (HD/CS); Deterrence Operations; Major Combat Operations (MCOs); Irregular Warfare; Military Support to Stabilization Security, Transition, and Reconstruction Operations; Military Contribution to Cooperative Security. [60]

401	Homeland Defense and Civil Support (HD/CS)
402	Deterrence Operations
403	Major Combat Operations (MCO)
404	Irregular Warfare
405	Military Support to Stabilization Security, Transition, and Reconstruction Operations;
406	Military Contribution to Cooperative Security
407	Other Mission Area (specify in text extracted)

Table 11 - Core Mission Areas for Ontology Concept [58]

Table 11 shows the core mission areas available covered under the ontology concept design. Which Core Mission Areas are mentioned as relevant has been indicated in the document. For older documents, which describe mission areas differently, text in the "Other Mission Area" slot has been included.

5. Threat Context Data Cluster

Threat context captures the program's potential use case context. Programs are developed against adversaries' capabilities. Additionally, threats can change during the developmental cycle. Therefore, the threat contexts need to be identified early and reviewed throughout the lifecycle of the program. It is a critical feature of the ontology design to identify programs and systems to be used in a common threat context.

501	Qualitative description of mission and/or operational threat context
502	Reference to document providing intelligence estimates

Table 12 - Threat Context for Ontology Concept [58]

Table 12 covers the threat context areas covered as part of the ontology concept data slots.

6. Operational Concept Data Cluster

This slot helps determine how the candidate program is linked with the Joint Concepts (i.e., Joint Operational Concept (JOC) /Joint Functional Concept (JFC) /Joint Integration Concept (JIC)) and other standards. [61]

601	Service and Joint Concept (document citation)
602	Support for Strategic Analysis (SSA) product (document citation)
603	Other operational concept (document citation)

Table 13 - Operational Concepts for Ontology Concept [58]

Table 13 above covers the operational concept type included as part of the ontology concept design.

7. Prior Documents Data Cluster

701	CBA (document citation)
702	ICD (document citation)
703	CDD (document citation)
704	CPD (document citation)
705	MNS (document citation)
706	ORD (document citation)
707	AoA (document citation)
708	DPS Defense Planning Scenario (document citation)
709	Other prior document (document citation)

Table 14 - Prior Documents for Ontology Concept [58]

Table 14 above covers the cluster as needed to capture references to prior documents related to the document being reviewed.

8. Strategic, Operational and Tactical Tasks Data Cluster

This cluster captures the UJTL task numbers appearing in the document. Also included are any Army, Navy, Marine Corps, or Air Force task numbers that appear. Tasks as they appeared in the document were not translated into current terminology. Also included were the conditions from the UJTL Conditions Taxonomy if they appeared in the document. If other types of capabilities tasks were mentioned, rows were replicated and included in the "Other task/activity" slot.

801	UJTL task (SN, ST, OP, TA level task numbers)
802	Army Task (ART task number)
803	Navy Task (NTT task number)
804	Marine Corps Task (MCT task number)
805	Air Force Task (AFSN, AFST, AFOP, AFTA, AFT task number)
806	Condition (as used, based on UJTL Conditions taxonomy)
807	Standards/Metrics
808	Other task/activity (description)

Table 15 - Strategic, Operational and Tactical Tasks for Ontology Concept [58]

Table 15 covers the strategic, operational and tactical tasks for ontology concept.

9. Document Level Joint Capability Areas Data Cluster

Joint Capability Area (JCA) is a collection of alike DoD capabilities functionally grouped to support capability analysis, strategy development, investment decision making, capability portfolio management, and capabilities-based force development and operational planning. JCAs are aligned with Functional Configuration Boards (FCBs). Currently, there are nine JCAs tier 1, aligned with FCBs as shown below in table 16: [62]

JCA	FCB Name
JCA #1 – Force Support (FS)	Force Support (FS)
JCA #2 – Battlespace Awareness (BA)	Battlespace Awareness
JCA #3 – Force Application (FA)	Force Application (FA)
JCA #4 – Logistics (LOG)	Logistics (LOG)
JCA #5 – Command and Control	Command, Control, Communications, and Computers /Cyber
JCA #6 – Communications and Computers	Command, Control, Communications, and Computers /Cyber
JCA #7 – Protection	Protection
JCA #8 – Building Partnerships	Force Support (FS)
JCA #9 – Corporate Management	Not assigned

Table 16 - JCA to FCB Mapping

901	Current JCA Tier 1 Name and/or Number
902	Current JCA Tier 2 Name and/or Number
903	Current JCA Tier 3 Name and/or Number
904	Current JCA Tier 4 Name and/or Number
905	Current JCA Tier 5 Name and/or Number
906	Current JCA Tier 6 Name and/or Number
907	Joint Functional Concept category name
908	FCB name (if specified)

Table 17 - Joint Capability Areas for Ontology Concept [58]

Table 17 above shows the six tiers of JCA being captured from the source document.

10. Capability Requirements Data Cluster

This cluster captures what the document says about capability requirements (not capability gaps recorded in section 5.4.12 below). If multiple capabilities requirements or if multiple attributes/metrics/values/threats are mentioned, each had been grouped together using Item Cross-Reference Tags to associate rows together as needed to capture document meaning.

1001	Description
1002	Capability attributes/metrics
1003	Capability attribute value - Objectives vs. metrics
1004	Capability attribute value - Objectives Time Frames
1005	Capability-specific threats and risks

Table 18 - Capability Requirements for Ontology Concept [58]

Table 18 above shows the capability requirements types captured under the ontology concept.

11. Legacy Fielded In-Development Capability Data Cluster

This cluster captures statements about legacy and in-development capability systems that the document mentions as related. Included are what the document says about the period of changes (positive or negative) in legacy and in-development systems and the anticipated changes to capability attribute values. Item Cross-Reference Tags have been used to associate rows together as needed to capture document meaning.

1101	Legacy capability system (name of system, or description)
1102	In-development capability system (name of system, or description)
1103	Time-frame of changes anticipated in legacy + development capability
1104	Capability attribute value changes anticipated over time

Table 19 - Legacy Fielded and In-Development Capability for Ontology Concept [58]

Table 19 above shows the legacy fielded and in development capability types.

12. Capability Gaps and Overlaps Data Cluster

The Capability Gaps cluster was the most complex to extract. In table 20, the orange color of the first line of a gap group of rows makes the group of rows easier to see during the extraction process. An extra empty row before each gap group of rows also helps.

The blank extraction form has groups for four gaps. Green Item Cross-Reference Tags are used to associate all the rows of each gap together. All rows for gap 1 had a green tag beginning with "G1", gap 2 beginning with "G2", etc. Rows (and groups of rows) were replicated as needed to capture the document content. For example, the JCA slots can be replicated and put in any order that is convenient. The first four rows (gap number, priority, time frame, and functional concept) generally occur once per gap.

Gap description and reason form a group of rows. If there is more than one reason, the reason row was replicated within its group. To associate the group of description/reason rows together, each row in a group had the same green Item Cross-Reference Tag (such as "G1-A", "G1-B", etc.).

Operational attribute and value also form a pair of rows. These rows can be replicated in pairs. Again, each row in a group had the same green Item Cross-Reference Tag (such as "G1-P1", "G1-P2", etc.).

1201	Gap number (in document)
1202	Priority
1203	Time frame of capability gap/overlap
1204	Functional concept
1205	Gap description
1206	Reason for capability gap: * proficiency * sufficiency * lack of fielded capability solution * replacement due to aging * policy limitations
1211	Tier 1 JCA related to gap
1212	Tier 2 JCA related to gap
1213	Tier 3 JCA related to gap
1214	Tier 4 JCA related to gap
1215	Tier 5 JCA related to gap
1216	Tier 6 JCA related to gap
1221	Operational attribute (associated with JCA)
1222	Operational attribute values

Table 20 - Capability Gaps and Overlaps for Ontology Concept [58]

Table 20 above covers the capability gaps and overlaps for the ontology concept design.

13. Recommendations for Analysis of Alternatives (AoA) Data Cluster

This section captures the recommendations for analysis of alternatives to the capability requirements being analyzed under the source document.

1301	Evolution of a previously fielded capability solution(s)
1302	Replacement or recapitalization of a previously fielded capability solution(s)
1303	Transformational capability solution(s)
1304	Technology considerations

Table 21 - Recommendations for AoA Ontology Concept [58]

Table 21 above shows the concepts for covering the recommendations for AoA. Rows in this cluster were replicated as needed to capture the types of recommendations for AoA provided in the document. Also rows were replicated as needed to include Technology Considerations mentioned.

14. Related Systems Data Cluster

This covers the type of related systems that are mentioned in the source document.

1401	Related system

Table 22 - Related Systems for Ontology Concept [58]

Table 22 above shows the related systems for ontology concept.

15. Reference List Data Cluster

Generally, a document has a Reference section in the back. Rows in this cluster were replicated as needed to capture the list of references as it appeared in the document.

1501	Reference

Table 23 - Reference List for Ontology Concept [58]

Table 23 above shows the ontology concept for the reference list of other documents referenced in the source document.

APPENDIX D – Snippet of Joint Capability Areas (JCA)

1.	Force Support	2.	Battlespace Awareness
1.1	**Force Management**	2.1	Planning & Direction
1.1.1	Global Force Management	2.1.1	Define and Prioritize Requirements (P&D)
1.1.1.1	Apportionment	2.1.2	Develop Strategies (P&D)
1.1.1.2	Assignment	2.1.3	Task and Monitor Resources (P&D)
1.1.1.3	Allocation	2.1.4	Evaluation (P&D)
1.1.1.4	Readiness Reporting	2.2	Collection
1.1.2	Force Configuration	2.2.1	Signals Collection
1.1.3	Global Posture Execution	2.2.1.1	Communications (SC)
1.2	**Force Preparation**	2.2.1.2	Electronic Emissions .(SC)
1.2.1	Training	2.2.1.3	Foreign Instrumentation (SC)
1.2.2	Exercising	2.2.1.4	Cyber Network (SC)
1.2.3	Educating	2.2.2	Imagery Collection
1.2.3.1	Professional Military Education	2.2.2.1	Electro-Optical (IC)
1.2.3.2	Civilian Education	2.2.2.2	Light Detection and Ranging (IC)
1.2.4	Doctrine	2.2.2.3	RADAR (IC)
1.2.5	Lessons Learned	2.2.2.4	Sonar (IC)
1.2.6	Concepts	2.2.2.5	Physical Environment (IC)
1.2.7	Experimentation	2.2.3	Measurements and Signatures Collection
1.3	**Human Capital Management**	2.2.3.1	Electro-Optical (MSC)
1.3.1	Personnel and Family Support	2.2.3.2	Radar (MSC)
1.3.1.1	Community Support	2.2.3.3	Geophysical (MSC)
1.3.1.2	Casualty Assistance	2.2.3.4	Radio-Frequency (MSC)
1.3.1.3	Wounded, ill and Injured Support	2.2.3.5	Chemical / Biological Materials (MSC)
1.3.1.4	Religious Affairs	2.2.3.6	Nuclear Radiation (MSC)
1.3.1.4.1	Religious Support	2.2.4	Human Based Collection
1.3.1.4.2	Religious Advisement	2.2.4.1	Human Intelligence (HBC)
1.3.2	Personnel Management	2.2.4.2	Counterintelligence (HBC)
1.3.2.1	Manning	2.2.4.3	Observation (HBC)
1.3.2.2	Compensation	2.2.4.4	Biometrics Data (HBC)
1.3.2.3	Disability Evaluation	2.2.4.5	Documents & Media (HBC)
1.3.2.4	Personnel Accountability	2.2.4.6	Socio-Cultural Data (HBC)
1.4	**Health Readiness**	2.3	Processing / Exploitation
1.4.1	Force Health Protection	2.3.1	Data Transformation (PE)
1.4.1.1	Joint Human Performance Enhancement	2.3.2	Information Categorization (PE)
1.4.1.2	Non-clinical Preventive Medicine / Health Surveillance	2.4	Analysis, Prediction and Production
1.4.1.3	Provide Public Health / Veterinary Services	2.4.1	Integration (AP)
1.4.1.4	Provide Global Patient Movement	2.4.2	Evaluation (AP)
1.4.1.5	Casualty Management	2.4.3	Interpretation (AP)
1.4.1.6	Provide a Healthy Fit Force	2.4.4	Prediction (AP)
1.4.1.7	Detainee Medical Care	2.4.5	Product Generation (AP)
1.4.2	Health Service Delivery	2.5	BA Dissemination and Relay
1.4.2.1	Define the Health Benefit	2.5.1	BA Data Transmission (IDD)
1.4.2.2	Clinical Preventive Medicine	2.5.2	BA Data Access (IDD)
1.4.2.3	Diagnosis		
1.4.2.4	Treatment		
1.4.2.5	Rehabilitation		
1.4.2.6	Re-integration		
1.4.3	Health System Support		

Figure 41 - Snippet of JCA Tier 1 and JCA Tier 2

3.	Force Application
3.1	**Maneuver**
3.1.1	Maneuver to Engage (MTE)
3.1.1.1	Air (MTE)
3.1.1.2	Space (MTE)
3.1.1.3	Land (MTE)
3.1.1.4	Maritime (MTE)
3.1.1.5	Underground (MTE)
3.1.1.6	Underwater (MTE)
3.1.1.7	Cyberspace (MTE)
3.1.2	Maneuver to Insert (MTI)
3.1.2.1	Air (MTI)
3.1.2.2	Space (MTI)
3.1.2.3	Land (MTI)
3.1.2.4	Maritime (MTI)
3.1.2.5	Underground (MTI)
3.1.2.6	Underwater (MTI)
3.1.2.7	Cyberspace (MTI)
3.1.3	Maneuver to Influence (MTInf
3.1.3.1	Air (MTInfl)
3.1.3.2	Space (MTInfl)
3.1.3.3	Land (MTInfl)
3.1.3.4	Maritime (MTInfl)
3.1.3.5	Underground (MTInfl)
3.1.3.6	Underwater (MTInfl)
3.1.3.7	Cyberspace (MTInfl)
3.1.4	Maneuver to Secure (MTS)
3.1.4.1	Air (MTS)
3.1.4.2	Space (MTS)
3.1.4.3	Land (MTS)
3.1.4.3.1	Populations (MTSL)
3.1.4.3.2	Infrastructure (MTSL)
3.1.4.3.3	Resources (MTSL)
3.1.4.4	Maritime (MTS)
3.1.4.5	Underground (MTS)
3.1.4.6	Underwater (MTS)
3.1.4.7	Cyberspace (MTS)

3.2	**Engagement**
3.2.1	Kinetic Means
3.2.1.1	Fixed Target (EK)
3.2.1.1.1	Surface (EKF)
3.2.1.1.1.1	Point (EKFS)
3.2.1.1.1.1.1	Hardened (EKFSP)
3.2.1.1.1.1.2	Soft (EKFSP)
3.2.1.1.1.1.3	Chemical, Biological, Radiolo·
3.2.1.1.1.2	Area (EKFS)
3.2.1.1.1.2.1	Hardened (EKFSA)
3.2.1.1.1.2.2	Soft (EKFSA)
3.2.1.1.2	Underground (EKF)
3.2.1.1.2.1	Hardened (EKFU)
3.2.1.1.2.2	Chemical, Biological, Radiologic:
3.2.1.1.3	Underwater (EKF)
3.2.1.1.3.1	Surf Zone (EKFU)
3.2.1.1.3.2	Very Shallow (EKFU)
3.2.1.1.3.3	Shallow (EKFU)
3.2.1.1.3.4	Deep Water (EKFU)
3.2.1.2	Stationary Target (EK)
3.2.1.2.1	Surface (EKS)
3.2.1.2.1.1	Point (EKSS)
3.2.1.2.1.1.1	Hardened (EKSSP)
3.2.1.2.1.1.2	Soft (EKSSP)
3.2.1.2.1.1.3	Chemical, Biological, Radiolo·
3.2.1.2.1.2	Area (EKSS)
3.2.1.2.1.2.1	Hardened (EKSSA)
3.2.1.2.1.2.2	Soft (EKSSA)
3.2.1.2.2	Underground (EKS)
3.2.1.2.2.1	Sort (EKSU)
3.2.1.2.2.2	Chemical, Biological, Radiologic:
3.2.1.2.3	Underwater (EKS)
3.2.1.2.3.1	Surf Zone (EKSU)
3.2.1.2.3.2	Very Shallow (EKSU)
3.2.1.2.3.3	Shallow (EKSU)
3.2.1.2.3.4	Deep Water (EKSU)
3.2.1.3	Moving Targets (EK)
3.2.1.3.1	Air (EKM)
3.2.1.3.2	Space (EKM)
3.2.1.3.3	Surface (EKM)
3.2.1.3.3.1	Point (EKMS)
3.2.1.3.3.1.1	Hardened (EKMSP)
3.2.1.3.3.1.2	Soft (EKMSP)
3.2.1.3.3.1.3	Chemical, Biological, Radiolo·
3.2.1.3.3.2	Area (EKMS)
3.2.1.3.3.2.1	Hardened (EKMSA)
3.2.1.3.3.2.2	Soft (EKMSA)
3.2.1.3.4	Underground (EKM)
3.2.1.3.4.1	Soft (EKMU)
3.2.1.3.4.2	Chemical, Biological, Radiologic:
3.2.1.3.5	Underwater (EKM)
3.2.1.3.5.1	Surf Zone (EKMU)
3.2.1.3.5.2	Very Shallow (EKMU)
3.2.1.3.5.3	Shallow (EKMU)
3.2.1.3.5.4	Deep Water (EKMU)
3.2.2	Non-Kinetic Means
3.2.2.1	Fixed Target (ENK)
3.2.2.1.1	Surface (ENKF)
3.2.2.1.1.1	Point (ENKFS)
3.2.2.1.1.2	Area (ENKFS)

4.	Logistics
4.1	**Deployment and Distribution**
4.1.1	Move the Force
4.1.1.1	Strategically Move the Force
4.1.1.2	Operationally Move the Force
4.1.2	Sustain the Force
4.1.2.1	Deliver Non-Unit-Related Cargo
4.1.2.2	Deliver Non-Unit-Related Personnel
4.1.3	Operate the JDDE
4.2	**Supply**
4.2.1	Manage Supplies and Equipment
4.2.2	Inventory Management
4.2.3	Manage Global Supplier Networks
4.2.4	Assess Global Requirements, Resources, Capabilities and
4.2.5	Operate the Joint Supply Enterprise
4.3	**Maintain**
4.3.1	Depot Maintenance
4.3.1.1	Inspect
4.3.1.2	Test
4.3.1.3	Service
4.3.1.3.1	Activate / Inactivate
4.3.1.3.2	Reclaim
4.3.1.4	Repair
4.3.1.5	Rebuild
4.3.1.5.1	Modify
4.3.1.5.2	Renovate
4.3.1.6	Calibrate
4.3.2	Field Maintenance
4.3.2.1	Inspect
4.3.2.2	Test
4.3.2.3	Service
4.3.2.3.1	Activate / Inactivate
4.3.2.3.2	Reclaim
4.3.2.4	Repair
4.3.2.5	Rebuild
4.3.2	Modify
4.3.2.5.2	Renovate
4.3.2.6	Calibrate
4.4	**Logistic Services**
4.4.1	Food Service
4.4.1.1	Contingency Base Feeding
4.4.1.2	Forward Unit Feeding
4.4.1.3	Remote Unit Feeding
4.4.1.4	Installation Feeding
4.4.2	Water and Ice Service
4.4.2.1	Bulk Water (non-potable)
4.4.2.2	Bulk Water (potable)
4.4.2.3	Packaged Water (bottled/pouched)
4.4.2.4	Ice Service
4.4.3	Contingency Base Services
4.4.3.1	Shelter
4.4.3.2	Billeting
4.4.3.3	Utility Operations
4.4.3.4	Water Reuse
4.4.4	Hygiene Services
4.4.4.1	Personal Hygiene Services
4.4.4.2	Textile Services
4.4.5	Mortuary Affairs
4.5	**Operational Contract Support**
4.5.1	Contract Support Integration
4.5.2	Contractor Management

Figure 42 - Snippet of JCA Tier 3 and JCA Tier 4

5.	Command and Control
5.1	**Organize**
5.1.1	Establish and Maintain Unity of Effort with I
5.1.1.1	Cultivate Relations with Mission Partners
5.1.1.2	Cultivate Coordination with Partner Organiza
5.1.2	Structure Organization to Mission
5.1.2.1	Define Structure
5.1.2.2	Assess Capabilities
5.1.2.3	Assign Roles and Responsibilities
5.1.2.4	Integrate Capabilities
5.1.2.5	Establish Commanders' Expectations
5.1.3	Foster Organizational Collaboration
5.1.3.1	Establish Collaboration Policies
5.1.3.2	Establish Collaborative Procedures
5.2	**Understand**
5.2.1	Organize Information
5.2.1.1	Compile Information
5.2.1.2	Distill Information
5.2.1.3	Disseminate Information
5.2.2	Develop Knowledge and Situational Awarer
5.2.2.1	Understand Implications
5.2.2.2	Analyze Information
5.2.2.3	Define Knowledge Structure
5.2.3	Share Knowledge and Situational Awarenes
5.2.3.1	Define Associated Community
5.2.3.2	Establish Collective Meaning (Collaboration)
5.2.3.3	Prepare Distributable Context
5.3	**Planning**
5.3.1	Analyze problem
5.3.1.1	Analyze Situation
5.3.1.2	Document Problem Elements
5.3.2	Apply Situational Understanding
5.3.2.1	Evaluate Operational Environment
5.3.2.2	Determine Vulnerabilities
5.3.2.3	Determine Opportunities
5.3.3	Develop Strategy
5.3.3.1	Determine End State
5.3.3.2	Develop Assumptions
5.3.3.3	Develop Objectives
5.3.4	Develop Courses of Action
5.3.4.1	Assess Available Capabilities
5.3.4.2	Understand Objectives
5.3.4.3	Develop Options
5.3.5	Analyze Course of Action
5.3.5.1	Establish Selection Criteria
5.3.5.2	Evaluate Courses of Actions
5.4	**Decide**
5.4.1	Manage Risk
5.4.2	Select Actions
5.4.2.1	Select Course of Action
5.4.2.2	Select Plan
5.4.3	Establish Rule Sets
5.4.4	Establish Intent and Guidance
5.4.5	Intuit
5.5	**Direct**
5.5.1	Communicate Intent and Guidance
5.5.1.1	Issue Estimates
5.5.1.2	Issue Priorities
5.5.1.3	Issue Rule Sets
5.5.1.4	Provide CONOPS
5.5.1.5	Provide Warnings

6.	Net-Centric
6.1	**Information Transport (IT)**
6.1.1	Wired Transmission
6.1.1.1	Localized Communications
6.1.1.2	Long-Haul Telecommunications
6.1.2	Wireless Transmission
6.1.2.1	Line of Sight
6.1.2.2	Beyond Line of Sight
6.1.3	Switching and Routing
6.1.3.1	Communication Bridge
6.1.3.2	Communication Gateway
6.2	**Enterprise Services (ES)**
6.2.1	Information Sharing
6.2.2	Computing Services
6.2.2.1	Shared Computing
6.2.2.2	Distributed Computing
6.2.2.3	Server Services
6.2.2.4	End User Services
6.2.3	Core Enterprise Services
6.2.3.1	User Access (Portal)
6.2.3.2	Collaboration
6.2.3.3	Content Discovery
6.2.3.4	Content Delivery
6.2.3.5	Common Identity Assurance Services
6.2.3.6	Enterprise Messaging
6.2.3.7	Directory Services
6.2.3.8	Enterprise Application Software
6.2.4	Position, Navigation, and Timing (PNT)
6.2.4.1	Provide PNT Information
6.2.4.2	Utilize PNT Information
6.3	**Net Management**
6.3.1	Optimized Network Functions and Resources
6.3.1.1	Network Resource Visibility
6.3.1.2	Rapid Configuration Change
6.3.2	Deployable Scalable and Modular Networks
6.3.3	Spectrum Management
6.3.3.1	Spectrum Monitoring
6.3.3.2	Spectrum Assignment
6.3.3.3	Spectrum Deconfliction
6.3.4	Cyber Management
6.4	**Information Assurance**
6.4.1	Secure Information Exchange
6.4.1.1	Assure Access
6.4.1.2	Assure Transfer
6.4.2	Protect Data and Networks
6.4.2.1	Protect Against Network Infiltration
6.4.2.2	Protect Against Denial or Degradation of Services
6.4.2.3	Protect Against Disclosure or Modification of Data
6.4.3	Respond to Attack / Event
6.4.3.1	Detect Events
6.4.3.2	Analyze Events
6.4.3.3	Respond to Incidents

7.	Protection
7.1	**Prevent**
7.1.1	Prevent Kinetic Attack
7.1.1.1	Above (PK)
7.1.1.1.1	Maneuvering (PKA)
7.1.1.1.2	Non-Maneuvering (PKA)
7.1.1.2	Surface (PK)
7.1.1.2.1	Maneuvering (PKS)
7.1.1.2.2	Non-Maneuvering (PKS)
7.1.1.3	Sub-surface Kinetic (PK)
7.1.1.3.1	Maneuvering (PKSS)
7.1.1.3.2	Non-Maneuvering (PKSS)
7.1.2	Prevent Non-kinetic Attack
7.1.2.1	Above Surface (PN)
7.1.2.2	Surface (PN)
7.1.2.3	Sub-Surface (PN)
7.2	**Mitigate**
7.2.1	Mitigate Lethal Effects
7.2.1.1	Chemical (ML)
7.2.1.2	Biological (ML)
7.2.1.2.1	Contagious (MLB)
7.2.1.2.2	Non-contagious (MLB)
7.2.1.3	Radiological (ML)
7.2.1.4	Nuclear (ML)
7.2.1.5	Electro Magnetic Pulse (ML)
7.2.1.6	Explosives (ML)
7.2.1.7	Projectiles (ML)
7.2.1.8	Directed Energy (ML)
7.2.1.9	Natural Hazards (ML)
7.2.2	Mitigate Non-lethal Effects
7.2.2.1	Chemical (MN)
7.2.2.2	Biological (MN)
7.2.2.2.1	Contagious (MNB)
7.2.2.2.2	Non-contagious (MNB)
7.2.2.3	Radiological (MN)
7.2.2.4	Electro Magnetic Pulse (MN)
7.2.2.5	Explosives (MN)
7.2.2.6	Projectiles (MN)
7.2.2.7	Directed Energy (MN)
7.2.2.8	Electro-Magnetic Spectrum (MN)
7.2.2.9	Natural Hazards (MN)

Figure 43 - Snippet of JCA Tier 5, JCA Tier, and JCA Tier 7

8.	Building Partnerships
8.1	**Communicate**
8.1.1	Inform Domestic and Foreign Audiences
8.1.1.1	Develop Objective Information
8.1.1.2	Identify Misinformation and Disinformation
8.1.1.3	Deliver and Adjust Information
8.1.2	Persuade Partner Audiences
8.1.2.1	Identify Foreign Audience Attitudes
8.1.2.2	Develop Cognitive Programs and Products
8.1.2.3	Deliver and Adjust Persuasive Content
8.1.3	Influence Adversary and Competitor Audiences
8.1.3.1	Identify and Understand Adversary and Competitor Attitudes
8.1.3.2	Develop Influential Programs and Products
8.1.3.3	Deliver and Adjust Influential Content
8.2	**Shape**
8.2.1	Partner with Governments and Institutions
8.2.1.1	Engage Partners
8.2.1.2	Develop Partnership Agreements
8.2.1.3	Enhanced Language, Regional Expertise, and Culture (LREC)
8.2.2	Provide Aid to Foreign Partners and Institutions
8.2.2.1	Identify Aid Requirements
8.2.2.2	Supply Partner Aid
8.2.3	Build Capabilities and Capacities of Partners and Instituti
8.2.3.1	Determine Partner Requirements
8.2.3.2	Enhance Partner Capabilities and Capacities
8.2.4	Leverage Capacities and Capabilities of Security Estabs
8.2.4.1	Identify Foreign Security-Related Capabilities
8.2.4.2	Determine Utility of Foreign Security-Related Capabilities
8.2.4.3	Stimulate the Use of Foreign Security-Related Capabilities
8.2.5	Strengthen Global Defense Posture

9.	Corporate Management and S
9.1	**Advisory and Compliance**
9.1.1	Advice and External Matters
9.1.1.1	Legal Matters
9.1.1.2	Legislative Matters
9.1.2	Audit, Inspection and Investigation
9.1.2.1	Audits
9.1.2.2	Inspections
9.1.2.3	Investigations
9.1.3	Operational Test and Evaluation
9.2	**Strategy and Assessment**
9.2.1	Strategy Development
9.2.2	Capabilities Development
9.2.3	Enterprise-Wide Assessment
9.2.4	Studies and Analyses
9.2.5	Enterprise Architecture
9.3	**Information Management**
9.4	**Acquisition & Technology**
9.4.1	Research
9.4.1.1	Basic
9.4.1.2	Applied
9.4.2	Advanced Technology
9.4.2.1	Capability Experimentation
9.4.2.2	Capability Demonstration
9.4.3	Developmental Engineering
9.4.3.1	Systems Engineering & Manufacturing
9.4.3.2	Developmental Testing
9.4.4	Acquisition
9.4.4.1	Program Initiation
9.4.4.2	Contracting
9.4.4.3	Portfolio System Acquisition
9.4.4.4	Production & Lifecycle Acquisition
9.4.4.5	Capability Termination & Disposal
9.5	**Program, Budget and Finance**
9.5.1	Program / Budget and Performance
9.5.2	Accounting and Finance

Figure 44 - Snippet of JCA Tier 8 and JCA Tier 9

Bibliography

[1] F. Kendall, "Performance of the Defense Acquisition System, 2013 Annual Report," Office of the Under Secretary of Defense, Acquisition, Technology and Logistics, 2013, Washington, DC, 2013. Available: https://www.defense.gov/Portals/1/Documents/pubs/PerformanceoftheDefenseAcquisitionSystem-2013AnnualReport.pdf [Accessed 06 January 2018].

[2] F. Kendall, "Performance of the Defense Acquisition System, 2016 Annual Report," Under Secretary of Defense, Acquisition, Technology, and Logistics. (2016). Department of Defense., Washington, DC, 2016. Available: https://www.defense.gov/Portals/1/Documents/pubs/Performance-of-Defense-Acquisition-System-2016.pdf [Accessed 06 January 2018].

[3] J. R. Enos, "Synthesizing DoDAF Architectures to Develop the Joint Capability Enterprise Architecture," in *Systems Engineering Conference*, Washington DC, 2014. Available: http://www.sedcconference.org/synthesizing-dodaf-architectures-to-develop-the-joint-capability-enterprise-architecture [Accessed 06 January 2018].

[4] E. Brynjolfsson and K. Mcelheran, "The Rapid Adoption of Data-Driven Decision-Making," *American Economic Review*, vol. 106, no. 5, pp. 133-139, 2016. Available: http://dx.doi.org/10.1257/aer.p20161016 [Accessed 06 January 2018].

[5] GlobalSecurity.org, "Joint Common Missile," [Online]. Available: https://www.globalsecurity.org/military/systems/munitions/cm.htm. [Accessed 06 January 2018].

[6] Defense Update, "JCM - Joint Common Missile," [Online]. Available: http://defense-update.com/products/j/jcm.htm. [Accessed 06 January 2018].

[7] Defense Industry Daily, "JAGM: Joint Air-Ground Missile Again," Defense Industry Daily, 06 June 2016. [Online]. Available: https://www.defenseindustrydaily.com/joint-common-missile-program-fired-but-not-forgotten-0229/. [Accessed 02 January 2018].

[8] Inspector General, Department of Defense, "Joint Air-to-Ground Missile Program," Alexandria, VA, 2017. Available: https://media.defense.gov/2017/Dec/08/2001853957/-1/-1/1/DODIG-2018-038_REDACTED.PDF [Accessed 06 January 2018].

[9] Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, [Online]. Available: https://www.acq.osd.mil/fo/images/atlSeal-SMALL.png. [Accessed 03 January 2018].

[10] Program Executive Office Missiles & Space, [Online]. Available: https://www.msl.army.mil/. [Accessed 03 January 2018].

[11] Naval Air Systems Command, [Online]. Available: http://www.navair.navy.mil/index.cfm?fuseaction=home.display&key=DBBA30A0-1AD2-4F26-BC09-B9FBF6661428. [Accessed 03 January 2018].

[12] D. M. Tate, "Acquisition Cycle Time: Defining the Problem (Revised)," Institute for defense Analyses, Alexandria, VA, 2016. Available: https://www.ida.org/https://www.ida.org:443/idamedia/Corporate/Files/Publications/IDA_Documents/CARD/2016/D-5762.pdf [Accessed 06 January 2018].

[13] D. H. Rhodes, "Systems Engineering in the Enterprise," MIT, Cambridge, 2008.

[14] T. Allen, D. Nightingale and E. Murman, "Engineering Systems: An Enterprise Perspective," in *Engineering Systems Symposium*, Cambridge, 2004.

[15] Wikipedia, "Joint Capabilities Integration Development System," Wikimedia, 29 November 2017. [Online]. Available: https://en.wikipedia.org/wiki/Joint_Capabilities_Integration_Development_System. [Accessed 03 January 2018].

[16] N. Ahmed, Improving Trade Visibility and Fidelity in Defense Requirements Portfolio Management, Cambridge, MA: Massachusetts Institute of Technology, 2014. Available: http://library.mit.edu/item/002522569 [Accessed 06 January 2018].

[17] AcqNotes, "JCIDS Process Overview," AcqNotes, 2017. [Online]. Available: http://acqnotes.com/acqnote/acquisitions/jcids-overview. [Accessed 25 December 2017].

[18] AcqNotes, "Capabilities Based Assessment (CBA)," AcqNotes, 2017. [Online]. Available: http://acqnotes.com/acqnote/acquisitions/capabilities-based-assessment-cba. [Accessed 25 December 2017].

[19] AcqNotes, "Initial Capabilities Document (ICD)," AcqNotes, 2017. [Online]. Available: http://acqnotes.com/acqnote/acquisitions/initial-capabilities-document-icd. [Accessed 25 December 2017].

[20] AcqNotes, "Capability Development Document (CDD)," AcqNotes, 2017. [Online]. Available: http://acqnotes.com/acqnote/acquisitions/capability-development-document-cdd. [Accessed 25 December 2017].

[21] AcqNotes, "Capability Production Document (CPD)," AcqNotes, 2017. [Online]. Available: http://acqnotes.com/acqnote/acquisitions/capability-production-document. [Accessed 25 December 2017].

[22] Joint Staff, "Manual for the Operation of the Joint Capabilities Integration," JCIDS Manual, 2012. Available:

http://www.jcs.mil/Portals/36/Documents/Library/Instructions/3170_01a.pdf?ver=2016-02-05-175022-720 [Accessed 06 January 2018].

[23] Joint Cheifs of Staff, "Chairman of the Joint Cheifs of Staff Instruction (CJCSI 5123.01G)," 2015. Available: http://www.jcs.mil/Portals/36/Documents/Library/Instructions/5123_01.pdf?ver=2016-02-05-175042-203 [Accessed 06 January 2018].

[24] Defense Acquisition University, "CJCS Diagram Capability Mission Lattice Legal Size," 27 July 2015. [Online]. Available: https://www.dau.mil/cop/rqmt/DAU%20Sponsored%20Documents/CJCS%20Diagram%20Capability%20Mission%20Lattice%20Legal%20Size%2027%20Jul%202015.pdf. [Accessed 04 January 2018].

[25] A. Moulton, D. Rhodes, S. Madnick, J. Enos and D. Matty, "Applying Semantic Technology to Early Stage Defense Capability Planning Analysis Based on JCIDS Artifacts," in *18th NDIA Systems Engineering Conference*, Springfield, VA, 2015. Available: https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2015/system/18026_Moulton.pdf [Accessed 06 January 2018].

[26] Wikipedia, "Semantic Technology," Wikimedia, 06 October 2016. [Online]. Available: https://en.wikipedia.org/wiki/Semantic_technology. [Accessed 25 December 2017].

[27] D. Allemang and J. A. Hendler, Semantic Web for the Working Ontologist, Waltham, MA: Morgan Kaufmann/Elsevier, c2011, 2011. Available: https://search.ebscohost.com/login.aspx?direct=true&db=cat00916a&AN=mit.002007484&site=eds-live&scope=site [Accessed 06 January 2018].

[28] Cambridge Semantics, "Introduction to the Semantic Web," Cambridge Semantics, 2017. [Online]. Available: https://www.cambridgesemantics.com/blog/semantic-university/intro-semantic-web/intro-linked-data/. [Accessed 25 December 2017].

[29] Cambridge Semantics, "Semantic University," Cambridge Semantics, 2017. [Online]. Available: https://www.cambridgesemantics.com/blog/semantic-university/learn-rdf/. [Accessed 25 December 2017].

[30] The World Wide Web Consortium, "Resource Description Framework (RDF): Concepts and Abstract Syntax," W3C, 2004. [Online]. Available: https://www.w3.org/TR/rdf-concepts/. [Accessed 25 December 2017].

[31] The World Wide Web Consortium, "RDF 1.1 Primer," The World Wide Web Consortium, 2014. [Online]. Available: https://www.w3.org/TR/rdf11-primer/. [Accessed 25 December 2017].

[32] The World Wide Web Consortium, "RDF Schema 1.1," The World Wide Web Consortium, 2014. [Online]. Available: https://www.w3.org/TR/rdf-schema/. [Accessed 25 December 2017].

[33] The World Wide Web Consortium, "OWL Web Ontology Language Overview," The World Wide Web Consortium, 2004. [Online]. Available: https://www.w3.org/TR/owl-features/. [Accessed 25 December 2017].

[34] Wikipedia, "Description Logic," Wikimedia, [Online]. Available: https://en.wikipedia.org/wiki/Description_logic. [Accessed 06 January 2018].

[35] M. Schneider and G. Sutcliffe, "Reasoning in the OWL 2 Full Ontology Language using First-Order Automated Theorem Proving," *Cornell University Library,* p. 2, 31 July 2011. Available: https://arxiv.org/pdf/1108.0155.pdf [Accessed 06 January 2018].

[36] https://ontotext.com, https://ontotext.com, [Online]. Available: https://ontotext.com/knowledgehub/fundamentals/what-is-sparql/. [Accessed 25 December 2017].

[37] D. E. Hahn, Predicting Program Success - Not Child's Play, Defense AT&L, 2014, pp. 45-47.

[38] M. U. Kevin P. Sweeney, Miniature Munitions: Is The US Military Prepared To Support Major Combat Operations?, Maxwell Air Force Base, Alabama: Air Command and Saff College Air University, 2016. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/1041216.pdf [Accessed 06 January 2018].

[39] Air Force Magazine, "Joint Future Theater Lift (JFTL) Technology Study (JTS) | Capability Request for Information (CRFI)," Air Force Magazine, 20 October 2010. [Online]. Available: http://www.airforcemag.com/SiteCollectionDocuments/ Reports/2010/October%202010/Day25/JFTL_CRFI__102210.pdf. [Accessed 19 December 2017].

[40] L. F. Ilse, "Military Intelligence Professional Bulletin," Federation of American Scientists, October-December 2012. [Online]. Available: https://fas.org/irp/agency/army/mipb/2011_03.pdf. [Accessed 25 December 2017].

[41] Hexplane Corp, "Joint Future Theater Lift (JFTL)," [Online]. Available: http://www.hexplane.com/JHL.html. [Accessed 09 January 2018].

[42] Citizen Airmen Magazine, "ISR AIrmen Make the Difference," [Online]. Available: http://www.citamn.afrc.af.mil/News/Features/Article/676437/ isr-airmen-make-the-difference-people-not-superior-technology-and-equipment-the/. [Accessed 09 January 2018].

[43] Wikimedia, "Lockheed Martin F22 Raptor," Wikimedia, [Online]. Available: https://upload.wikimedia.org/wikipedia/ commons/thumb/4/46/ Lockheed_Martin_F-22A_Raptor_JSOH.jpg/ 1200px-Lockheed_Martin_F-22A_Raptor_JSOH.jpg. [Accessed 25 December 2017].

[44] https://static.interestingengineering.com, "F15 Eagle," https://static.interestingengineering.com, [Online]. Available: https://static.interestingengineering.com/ images/import/2017/06/F15Eagle.jpg. [Accessed 25 December 2017].

[45] Wikimedia, [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/ 5/51/Stryker_ICV_front_q.jpg. [Accessed 25 December 2017].

[46] Space Today Online, [Online]. Available: http://spacetoday.org/images/Sats/MilSats/GPSbig.jpg. [Accessed 25 December 2017].

[47] https://www.copybook.com, [Online]. Available: https://www.copybook.com/media/military/ profiles/thalesraytheon-systems-trs/migrated/ images/thalesraytheon18.jpg. [Accessed 25 December 2017].

[48] The National Interest, 2017. [Online]. Available: http://nationalinterest.org/files/styles/main_image_on_posts/ public/main_images/f-16_fighting_falcon.jpg?itok=CBB4VmdL. [Accessed 25 December 2017].

[49] http://www.military-today.com, [Online]. Available: http://www.military-today.com/missiles/aim120_amraam.jpg. [Accessed 25 December 2017].

[50] United States Air Force, 2017. [Online]. Available: http://www.af.mil/About-Us/Fact-Sheets/Display/Article/ 104572/joint-direct-attack-munition-gbu-313238/. [Accessed 22 December 2017].

[51] United States Air Force, [Online]. Available: http://www.af.mil/About-Us/Fact-Sheets/Display/Article/104505/f-16-fighting-falcon/. [Accessed 25 December 2017].

[52] United States Air Force, [Online]. Available: http://www.af.mil/About-Us/Fact-Sheets/Display/ Article/104576/aim-120-amraam/]. [Accessed 25 December 2017].

[53] L. Bottou, "From Machine Learning to Machine Reasoning," 2011. Available: https://arxiv.org/abs/1102.1808 [Accessed 06 January 2018].

[54] Apache Drools, [Online]. Available: https://www.drools.org/. [Accessed 25 December 2017].

[55] Apache Jena, [Online]. Available: https://jena.apache.org/documentation/inference/. [Accessed 25 December 2017].

[56] GitHub, [Online]. Available: https://github.com/stardog-union/pellet. [Accessed 25 December 2017].

[57] S. Robertson and J. Robertson, "Mastering the Requirements Process," Atlantic Systems Guild Ltd, [Online]. Available: http://www.volere.co.uk/masteringrequirementsprocess.htm.

[58] Allen Moulton, "JCIDS Semantic Architecture Framework Study Guidance on Manual Text Extraction Form & Process," 2014.

[59] Defense Acquisition University, "DoD Strategic Guidance," Defense Acquisition University, 2017. [Online]. Available: https://www.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=45b89576-2063-4ef9-87d3-f85a2942ea9a. [Accessed 25 December 2017].

[60] Defense Acquisition University, "Core Mission Area-Defense Acquisition Glossary(DAP)," Defense Acquisition University, 2017. [Online]. Available: https://www.dau.mil/glossary/pages/3016.aspx. [Accessed 25 December 2017].

[61] K. J. Baldwin, "Systems and Software Engineering, Defense Acquisition Program Support Methodology," Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, 2009. [Online]. Available: https://www.acq.osd.mil/se/docs/DAPS_V2.0_Methodology.pdf. [Accessed 25 December 2017].

[62] Defense Acquisition University, "Joint Capability Area (JCA) - Defense Acquisition Glossary (DAP)," Defense Acquisition University, 2017. [Online]. Available: https://www.dau.mil/glossary/pages/2104.aspx. [Accessed 25 December 2017].

[63] Joint Electronic Library, "Joint Electronic Library, UJTL Tasks," Joint Electronic Library, 2017. [Online]. Available: http://www.dtic.mil/doctrine/training/ujtl_tasks.htm. [Accessed 25 December 2017].