13th ICCRTS: C2 for Complex Endeavors

"Managing the Life Cycle of Net-Centric Architectures"

Topic 8: C2 Architectures

Name of Author(s): John V. Tieso, Charles Thornburg

Point of Contact: John V. Tieso

Name of Organization: Booz Allen Hamilton

1550 Crystal Drive, Suite 1100

Arlington VA 22202-4158

Telephone; 703-412-7464

E-mail Address; Tieso_John@bah.com

Title of Paper: "Managing the Life Cycle of Net-Centric Architectures"

Abstract

Information architecture development has been an integral part of the command and control community for many years, often as an exercise for satisfying legal, regulatory, or internal requirements to 'justify' requirements. The development of architectures has a broader meaning, and a critical purpose, especially in a net-centric environment.

Over the past 8-10 years the question of net-centricity, and its relationship to command and control requirements, especially those involving increasingly integrated force execution through joint task forces, and other joint operations, have also increased the need for utilizing lessons learned in previous operations. Many of these operations or actions were documented in architectures, and the information described in these architectures has value, through reuse, in future operations.

This paper discusses the life cycle of information architectures, and the concurrent need to organize, sort, maintain, review, and prepare architecture information for reuse. It shows how the data, organized as information, can be applied to other, similar circumstances when required. The paper also points out how frameworks for organizing data, such as the DoD Architecture Framework (DoDAF), the NATO Architecture Framework (NAF), and the Ministry of Defense Architecture Framework (MODAF) can contribute to coalition and multi-national solutions by reuse of existing architectural data.

Keywords: Architecture, Enterprise Architecture, Coalition Architecture, Data, Information, Information Architecture, Reuse, Net-Centric

Introduction

The development of information architectures to frame process improvement, systems development, or information technology deployment is a well-established technique for focusing change efforts within an organization. In the commercial world, architecture methods and techniques, such as that developed by John Zachman[1], originally with International Business machines, Inc. (IBM) have existed since the late 1970's. Early architecture efforts were primarily focused on systems engineering and development, while corresponding modeling and simulation projects, led by industry pioneers, such as R. Douglas Ross of MIT and Softech Inc, Robert Shapiro of META Software, and Dan Appleton of DACOM Inc., produced tools and techniques for documenting business processes and their associated business rules. However, significant disconnects in the industry delayed the ultimate consolidation of many tools and techniques into a consolidated workbench which could efficiently bridge the gap between process understanding and systems development until the mid-1990's.

Net-Centric Requirements and Operations as a Development Factor

The U. S. Department of Defense (DoD) can easily be considered one of the largest developers and users of networked services in the world. The Department has been concerned with that requirement for many years, and has executed several initiatives, starting in the late 1980s, and continuing today. The Corporate Information Management (CIM) Initiative developed the Technical Architecture for Information Management (TAFIM) and the Joint Technical Architecture (JTA), both designed to foster and enhance networked interoperability. These efforts were later incorporated in The Open Group Architecture Framework (TOGAF) developed for commercial use by the Object Management Group (OMG) as a commercial standard.

During the mid-1990's, DoD began executing three major information-related initiatives evolving in virtually parallel environments. The first, development of a new framework for architecture-based management of operations, acquisition, and technology to replace the TAFIM, was begun in 1995 in the office of the Assistant Secretary of Defense (Command, Control, Communications, and Intelligence) (C3I)[2]. This framework, the Command, Control, Communications, Computing, Intelligence, Security & Reconnaissance (C4ISR) Architecture Framework, Version 1, was published in 1996 as a means of better defining interoperability and integration among systems development effort in the command and control communities of interest (COI) of the DoD. The C4ISR Framework described methods and techniques for creation of information architectures supporting command and control-related systems engineering activities.

On a second front, a major effort was made to begin the long process of defining network-centric operations, and the support required within the Department to achieve that objective. The invasion of Kuwait by Iraq in 1990 (Iraq 1) intensified the concerns of DoD that it lacked the capability to respond with significant force in response to threat to either itself or its allies. Traditional methods of data gathering and analysis took too long, and the press of modern warfare, together with the increasing importance of network-based command and control, made it critical that the DoD change its methods of information delivery and analysis. Lessons learned carefully documented in the theatre, and within the combatant commands reinforced the demonstrated need for change. Importantly, DoD listened to those who were experiencing the conflict first hand in Iraq1, rather than listening solely to the academics and others who had not experienced conflict. Increasingly, the message became clear—the future depended on available, efficient, networked-based communications driven over both traditional wired circuits, and over the increasingly useful Internet.

The Global Information Grid (GIG) Architecture was created to describe net-centric operations, and establish an enterprise context within which exists the description of current IT capabilities and its supporting environment, and the design for the future networked DoD operational environment. In

February, 2003, development of the Net-centric Operations & Warfare Reference Model (NCOW-RM) was announced, which would describe the net-centric environment, to include services, capabilities, the infrastructure supporting the GIG, and the standards and technologies needed to create next generation net-centric operations. Capping the GIG initiative was the development of a federation strategy, through which DoD organizations would be able to develop and maintain their applications, systems, and networks while assuring interoperability through adherence to GIG/NCOW principles and practices.

The third major initiative responded to the passage of the Clinger-Cohen amendments to the 1996 Defense Appropriations Act[3], which required the use of information architecture views in support of appropriation requests for major governmental systems as a means of determining the expected use of the requested system. That initiative involved the rewriting of the C4ISR Framework to make the framework applicable department-wide for business operations, and was published as C4ISR Framework Version 2.0 in 1997. The document was eventually renamed the DoD Architecture Framework (DoDAF), Version 1.0, and published in August, 2003 by the office of the Assistant Secretary of Defense (Network Interoperability & Integration) (ASD (NII)), the successor to ASD (C3I). In turn, the DoDAF has gone through several changes and updates over the past several years, and is presently being revised as DoDAF, Version 2.0.

Joint Vision 2010 and Joint Vision 2020 articulated the comprehensive vision of net-centric operations. Publication of DoDAF 1.0 was one of the first steps facilitating the ability to move forward with a transformation policy in the Department to achieve the comprehensive net-centric operations vision. The GIG provides the structure, the fabric, on which to define and create net-centric technology and infrastructure capabilities. The NCOW-RM, the reference model, provides the 'what'--the descriptions of the needed parts and supporting mechanisms that would comprise the net-centric environment, and the DoDAF provides a means to collect and organize information required within the net-centric environment, present it graphically and in native data form, and support development of net-centric operations and technology needed within the environment.

The earlier versions of DoDAF concentrated on describing and defining an understandable set of products (called 'views') which graphically presented information on an operation or system undergoing analysis for some purpose. However, two significant changes are occurring in both Government and the commercial space that is dramatically affecting the whole practice of architecture. First, the sense of architecture as a set of views is changing, and giving way to a new sense of the important of architecture development as a *data-centric activity*, rather than as a *product or view-centered activity*. Second, the concept of net-centricity is affecting the way business is conducted in an even more dramatic way. The fact that good data drives most business operations (To include tactical or command

and control operations in the military) has been known for a very long time. However, the collection of data has not, over the years, proven to be a consistent or even desirable task, and was usually left to the technocrats who would collect, organize, assimilate, and analyze the data to death before trying to use it in real-time situations. Often, the analysis would prove that the wrong data had been collected and was, therefore, useless for executing decisions.

Architectures by their nature define processes and/or systems that support those processes. Applying the concept of data-centricity to process description, as we will show below, increases exponentially the power and value of the resulting architecture. In turn, when net-centricity is applied to enabling technology to facilitate the passage of data in some context, creating information, the technology supporting all types of operations further enhances the usefulness of a well-designed and documented architecture in the performance of real-time problems.

Finally, in the real world of tactical operations, which we use frequently as examples below, there are really three aspects to powerful architectures. The first two, data-centricity, and net-centricity, we have already briefly introduced. The third, *information validity*, is an overarching aspect in that it recognizes the inherent fragility of data, and its loss of value for creating information over time. All three aspects, working together in a life cycle, represent the main thrust of this paper.

The Enterprise Architecture Life Cycle

The principal thrust of this paper is to establish that a life cycle exists for information architectures. The life cycle involves both a *timeline* and a *link to a method* for creating and maintaining architectures. Both work in a synchronous way to ensure that an architecture remains valid, relevant, and current. This part of the paper builds a life cycle model, which starts with identification of a requirement. We link the architecture life cycle to critical net-centric Operations concepts, both as examples, and to further the understanding of the use of 'living' architecture in net-centric operations.

The NCOW Reference Model (NCOW-RM)[4] defines three operational models for net-centric operations; Evolve, Operate, and Maintain. Within these models are found architecture development (Evolve), maintenance, storage, and update (Use), and configuration management and review (Manage). A number of frameworks, reference models, policies and procedures interrelate with the NCOW-RMs, as shown in Table 1, below.

Document/ Service	EVOLVE	OPERATE	MANAGE	Comments
Architecture Framework	DODAF MODAF NAF TOGAF	TOGAF	TOGAF	DoDAF defines the mechanics for architecture data collection and architecture view/product creation
GIG	GIG, V.1 GIG V.2	GIG V.1 GIG V.2	GIG Architecture Federation Strategy	The GIG provides the enterprise context for architecture in the NCE
Net-centric Environment (NCE)	NCE	NCE	NCE	The framework for net-centric capabilities
GIG Enterprise Services Strategy	CES COI	CES COI		Focuses on services loosely coupled to systems
DoD Data Strategy	Net-centric Data Strategy (NCDS)	Net-centric Data Strategy	Net-centric Data Strategy	NCDS describes the vision for managing & sharing data in the NCE
DoD Services Strategy	DoD Net-Centric Services Strategy (NCSS)	DoD Net-Centric Services Strategy (NCSS)	DoD Net-Centric Services Strategy (NCSS)	NCSS
Information Assurance	DoD IA Strategy	DoD IA Strategy; IA Component of the GIG Architecture	DoD IA Strategy	Provides the basis for assured info capabilities in the NCE

Table 1 NCOW-RM relationships

There are major intersections among the various documents that together contribute to architecture development within the net-centric environment. DoDAF provides a means to ensure consistent architecture creation, maintenance, and reuse through adherence to its principles of data collection and product/view creation. DoDAF defines the architecture "language" for architects when collecting and sharing data, however, the products and views need to be presented in the "language" of the decision makers.[5] The GIG and NCOW-RM provide a means to consistently describe the major requirements of net-centricity, as they would contribute to an architecture effort. The DoD

Net-Centric Data Strategy amplifies on both the NCOW-RM and the DoDAF by providing a standard for data collection and, together with the DoD IA Strategy and the IA Component of the GIG Architecture, ensures that the results of the architecture effort are useful in real-time situations and problems that arise in Departmental operations. The GIG Architecture Federation Strategy ensures that the results of architecture development, the applications, systems, and network platforms are mutually supportive, where required, and support interoperability. How these interrelate in graphical terms is shown in Figure 1.

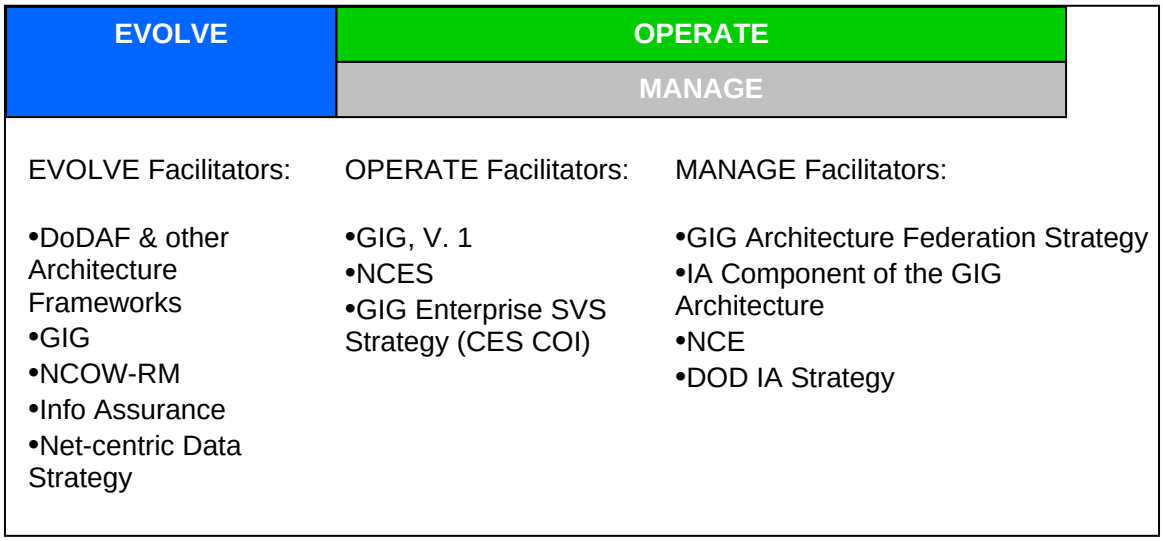

Figure 1. The Net-centric Architecture Life Cycle

It is this view we will carry forward in the rest of the paper.

Understanding requirements

Every architecture development project starts with a *requirement*. The requirement can be formal or informal, spoken or written, required or voluntary, personal or official, or for any purpose. Nonetheless, there is always some kind of requirement that drives the development of an architecture.

Requirements definition

Defining the requirement correctly is the first and most critical element of any successful project. This is also the case in defining a proposed architecture project. The first steps, then, should be to 'put the requirement to paper'. Quite often, a proposed project is not yet fully analyzed, and any requirement definition is subject to change. Nonetheless, attempt to describe the requirement as full as possible. That will allow initial planning and review, even if the requirement description is subject to change. Requirements are generally framed in terms of capabilities; standards

described by the NCOW-RM, and needed capabilities described by the proposing organization. When all of these enablers are considered together, the process of creating architectures to describe the requirements utilizing DoDAF data and views becomes much less burdensome.

Establishing an architecture vision

Figure 2 describes how a requirement transforms itself into an architecture project vision. A project vision is a simple statement of expected results desired to successfully execute a requirement. That statement may be limited or extended when utilizing guidance provided through an architecture development framework, such as DoDAF or TOGAF, or the prescriptions of the NCOW-RM.

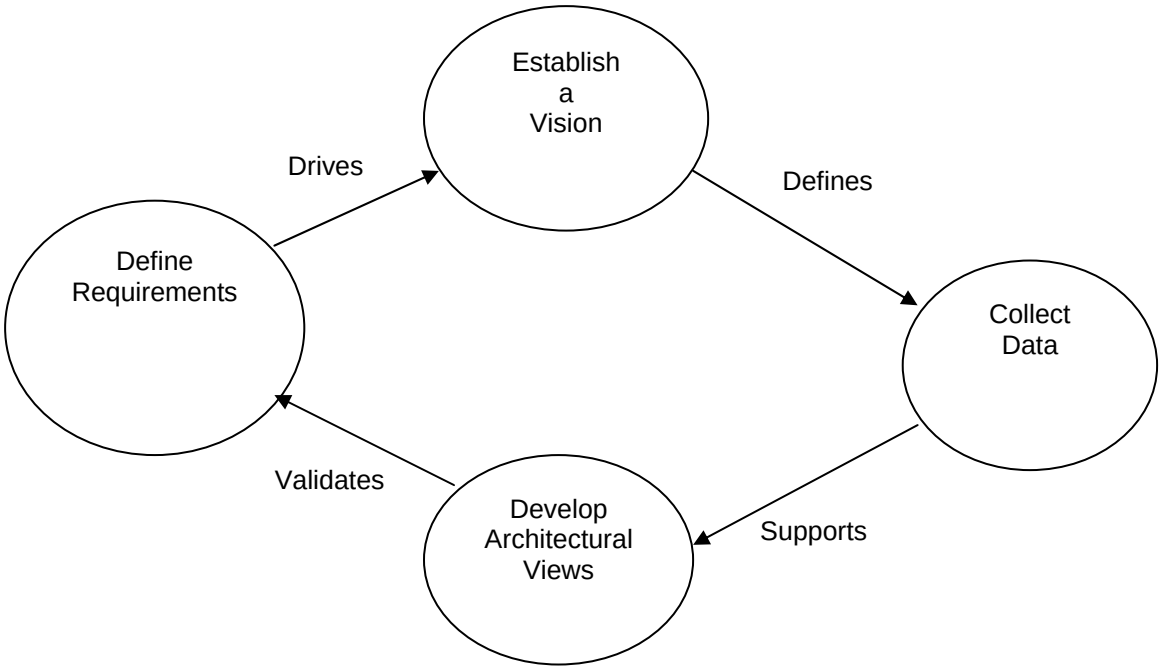

Figure 2. Requirements translate to vision

A vision, e.g., JV 2020, is an expected or desired outcome. When a vision is created of a future state, neither the process itself nor the supporting technology necessarily have to currently exist in any form. However, a vision, to be useful in a net-centric architecting process, must be measurable and it must relate to either an existing capability, or an expected future capability that will reside on, be transmitted by, and executed through some form of networked service.

The actions required for translating a vision into an executable process, and then maintaining that process over time is a bit more complex. First, the

vision must be able to state goals and objectives in terms that others can read and understand. In the DoD, the DoDAF serves that purpose, providing a means to understand how to collect data, and utilize that data in views which are useful to others in understanding what is expected from the process or operation. Using the DoDAF to guide development of the vision facilitates other actions as well, such as integration of the architecture into other architectural constructs, providing a means of integration with other processes that share space (i.e. an 'edge') with the architecture, and providing examples of views which can be used to represent the process or operations in graphical, easier-to-understand ways. According to DoD CIO/ASD (NII), the upcoming DoDAF 2.0 will have data and views that represent capabilities, strategies, acquisition needs, and human resource requirements, in addition to maintaining forward compatibility[6] with older versions of the Framework that continue to be used.

Data Collection

As seen in Figure 2, the vision defines what data needs to be collected and organized to achieve the desired result, and visualize that data through architectural views or products. The data needed is determined by the type of requirement, the standards or policies in place for data collection and utilization, and the expected outcome. Generally, data is collected in some form that facilitates its storage for later use in replicating or enhancing a previous outcome, and may also be subject to limitations imposed by data reference models, such as the NCOW-RM or the DoD EA RM, which prescribe the format for data needed for graphical presentation and reporting of results.

Creating integrated architecture views of data

Integrated architecture views, or products produced by collected data for presentation take many forms, but the views have several things in common regardless of the reason for their creation. These are:

- Data used to create the views are applied consistently, and with the same definition throughout the views without regard to the techniques or notation used

- Architecture design artifacts are either adopted from standard sources, or fully described in the manner suggested or required by the architecture framework being used for architecture development (i.e. DoDAF, MoDAF, NAF, TOGAF, etc.)

- Architecture products and views are created in the numbers and complexity necessary to support and fully explain the solution to the requirement.

- Where architecture views are necessary and other, previous efforts have produced views which touch the edge of a current view, then the

same data that created the original view should be used to create the new view. This creates automatic, integration of architecture views through reuse of already existing data, and also facilitates reference of the architecture to the higher level enterprise architecture.

Architecture views can be of two types; pre-defined views, such as those described in the DoDAF V1.5 and other major defense-oriented architecture frameworks, or ad-hoc views created using the data collected in response to a requirement, and in a form or format familiar to those needing or reviewing the data. Different levels of management and those executing the results of the architecture effort need views that are meaningful to their work.

As an example, creation of a system in response to a defined requirement for order of battle information needs to collect data that relates to organizations, roles, performance requirements, readiness, and available personnel. It needs to know what steps are involved in creating an order of battle scenario (i.e. the process), how that process is executed, and the expected outcome, in order to begin to understand the design of a system. It does not necessarily care about future standards, or other 'to-be' information—only on the information needed, where it is located, how it is collected and organized, how it is analyzed, and what result form is expected. From that kind of information, the information engineer can work with the systems engineer to break the process down into modules that can be created as functions and procedures within a system. To some extent, this information can be formatted into views that are described in the DoDAF or another of the architecture frameworks. However, the information needed by the systems engineer does not fit neatly into the 'standard' views or products, but into specifications, e.g., SYSML, and spreadsheets of information needed to create schematics of how the program modules will work. So, while some views may be a structured format to ensure that appropriate data is collected for analysis, other views may be in more recent methods, such as Unified modeling Language (UML) which can more easily translate into system engineering specifications.

As is seen in Figure 3, the creation of integrated architecture views starts a chain of events that lead to execution of a requirement. First, the views are validated against established requirements, the requirement solution is executed, and other services updated as a result of the effort undertaken in the architecture development effort.

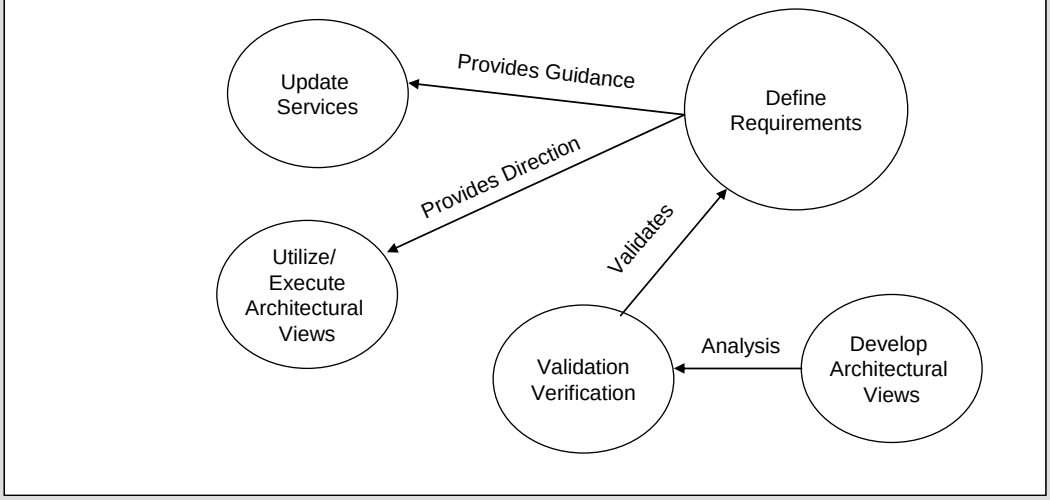

Figure 3 Architecture Execution

Validation and verification of architecture data

Once data is collected and architecture views developed, it is time to validate that the resulting drafts of the architecture views and the data actually meet the specifications contained in the original requirement statement. Validation steps need to occur fairly frequently to prevent a major inadvertent shift away from the desired or expected result into more unknown territory. Final verification of the data and views always occurs before the data is stored for future use after its immediate usage in executing the solution to a requirement is completed.

Storing & maintaining architecture data

Each of the major frameworks defines a technique and recommended notation for the storage of data collected during an architecture development effort. A critical part of the effort is determining how and where to store the collected data, in a format that can facilitate reuse by others with similar needs and requirements. Extensible Markup Language (XML) has increasingly taken over part of this responsibility as the format in which to transfer information to and from repositories regardless of repository database file structure. Similarly, data on the construction of architecture views can also be saved for reuse.

While early effort dwelled on the need for central repositories where data could be stored, this too is changing as federated approaches have become more common where the data reposes in repositories in virtually any location, and the central registry becomes more of a library system with

pointers to the data through internet/intranet links that can be accessed quickly from almost anywhere in the world or in the organization.

Whatever process is utilized for organizing and storing the information, some form of configuration management is required, as shown in Figure 4. Once the architecture views

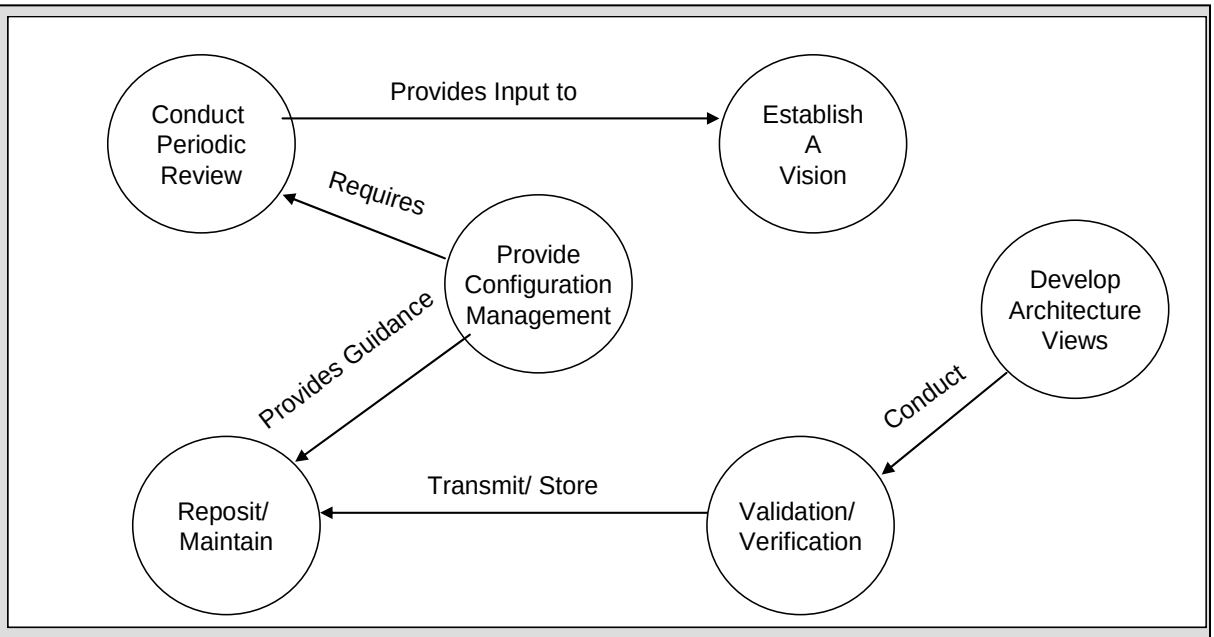

Figure 4 Managing Architecture Information

are developed, validated, and verified against a requirement, the data can be stored in a repository, and the rules of the organization applied to determine where and how it will be stored. A configuration management program determines the rules for these actions, and also facilitates later reuse through consistent application of those rules.

Reviewing architecture data

Periodic reviews of the data and the architecture artifacts (i.e. the views and products) will determine if the data is current and valid for use or should be updated or discarded. If further work is required, or a new requirement is created, then the information on the status of the data and artifacts will affect decisions to be made on the vision statement for executing development of the architecture to support that new requirement.

Reuse of architecture data

Finally, architecture data, properly collected, described, defined, and named according to the structures described in the DoDAF or other frameworks or

reference models is a prime candidate for reuse for later or updated requirements. Architecture data organized by community of interest (COI) or primary use can be searched, analyzed and adopted by others as a ready solution to the need to develop specific architectural or reference data. Utilization of the DoD XML Registry, the DoD Architecture Registry System (DARS), or other sources of authoritative data vetted through use, and under configuration management, is a 'first choice' selection when data is required that bears on the requirements process.

Applying Net-centric Principles to Architecture Development

Having now introduced at a high level the concept and steps of an architectural life cycle, it is time to turn to the application of Net-centric principles to architecture development in the Net-centric Environment (NCE).

The NCOW Reference Model defines three major models through which Net-centric operations are conceived, executed, and managed. These are shown below in figure 4, and are blocked out to indicate which parts of the architecture life cycle we have described above apply to those models.

Evolve the Architecture, the first model, involves creation of the vision as an expression of the requirement, and further definition of the vision through identification of data needed to create a desired solution that is responsive to the requirement. This solution includes all those operations, services, and systems necessary to fully execute the mission requirement. Each element of the proposed solution is represented by data which, in turn, is expressible through views or products produced conformant to the DoDAF or another architecture framework. These views are useful to the manager, systems engineer, information engineer, and others who use the data to perform work related to the design, execution, and management of the requirement. All data is validated and verified against the requirements and vision statement and desired outcomes.

Operate the Architecture, the second model, involves utilizing the results of validation and verification to execute the requirement, consistent with the requirements as described in the architecture data and views, maintaining the architecture data, and establishing configuration management and control for the data and information contained in the views.

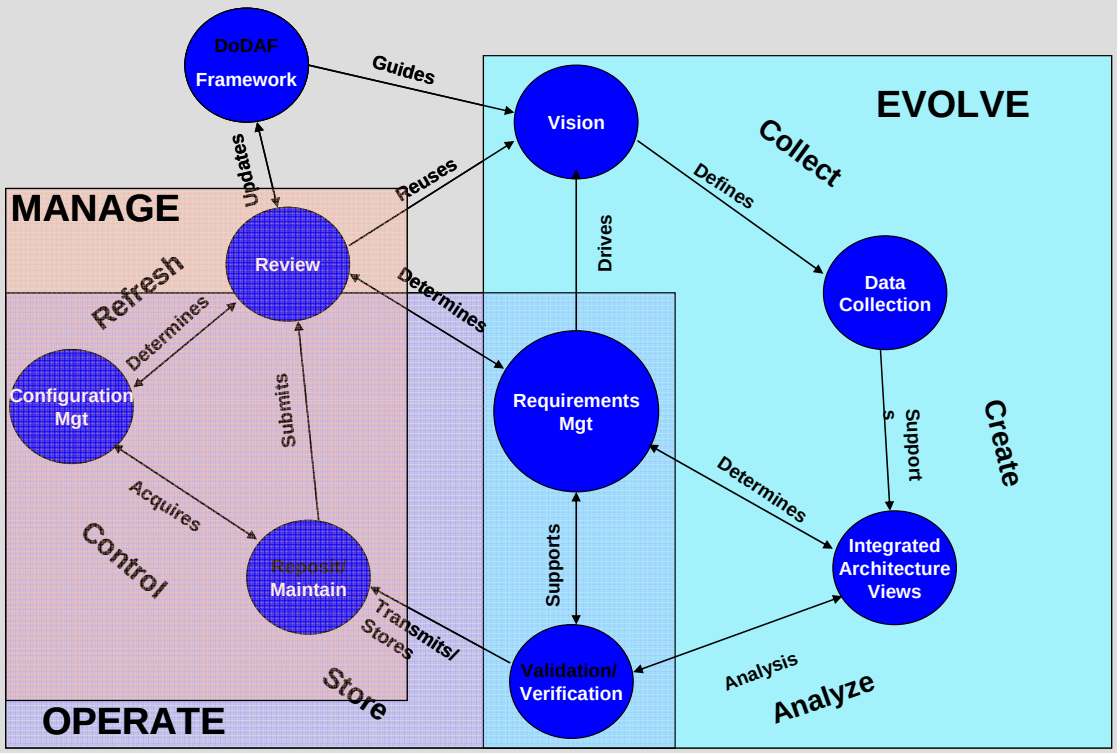

Figure 4 The Net-centric Architecture Life Cycle

Manage the Architecture utilizes the data contained in the repository under configuration control to conduct reviews on the validity, currency, and continued usefulness of the data, and its value for future efforts.

When developing a solution to a requirement under the Joint Capabilities Integration Development System (JCIDS) Process[7], as an example, a set of integrated architectures are utilized in the analysis process to determine at what level a needed capability exists, or must be developed, and current processes must be changed and/or upgraded to provide the needed solution. JCIDS defines a *Top-down Capability Need Identification Process* for that purpose. The life cycle described in Figure 4 supports that analysis from the requirement identification phase and the establishment of the capability vision to the collection of data needed to support the capability description. Importantly, data collection in this instance would include the data supporting both Joint Operating Concepts and Joint Functional Concepts, the architecture views that support both sets of concepts, and the creation of a consolidated or derived view of how those concepts would apply to the current requirement. These views are validated against the original requirement to ensure that the desired result is achieved.

Following analysis, using the data collected, and the derived views, a set of recommendations would follow, be validated against the stated required, and verification made that the needed instance data existed, and in what location. Once the recommendations had been made, and a decision reached, the architecture data is stored as reference data in the repository supporting the integrated architectures of record for future use, and periodically reviewed for continued validity and currency.

Early information architecture development concentrated primarily on processes and systems, making it very difficult to describe more complex requirements, such as capability development, which may rely in some instances on systems only peripherally. Rather, capability development depends more on *services* for its solutions to requirements. The evolution of services-oriented architectures (SOA), together with understanding on how this level of an architecture (Often acting as the intermediary between processes and systems) facilitates requirements analysis leading to development and execution of enhanced capabilities is rapidly becoming a mainstream approach. The approach needs to focus on provisioning of services in an integrated way that dramatically improves the ability to move data where decisions need to be made without complex analysis at each step.

The development of SOA and the net-centric principles reshaped the ability to analyze complex situations. By quickly focusing in on the location of most relevant data, SOA and the net-centric principles accomplish two major objectives:

- Establishing a way to visualize data through services and ad-hoc reports

- Gaining the ability to update data quickly through fast, efficient, and often decentralized data collection engines.

These changes in approach meant that the architectures supporting the changing or updating of capabilities to respond to new requirements had to change as well. The federated approach to architecture storage and maintenance means that architecture data for a particular purpose may come from many locations instead of a central location, and that ad-hoc reports may now be generated based on information passed through the secure Internet rather than slower communications routes. There may be several owners of information, giving rise to the need to determine and certify authoritative sources, where, in the past in-house data may have been used with less currency or validity than other sources.

Configuration Management plays a critical role in both the physical storage and maintenance of data, and ensuring its availability when needed. A Configuration Management Program provides rules for maintenance, update, and eventual replacement of data, and the mechanisms for ensuring that periodic reviews and updates occur that extends the period of validity and usefulness.

All of these changes require considerable preparation beyond what had been required in the past, even to produce architecture data and supporting views that limited to those described in the DoDAF, at least through version 1.5. The focus for DoDAF Version 2.0 places emphasis on the architecture data supporting multiple uses. "Fit for Purpose" is the approach using a framework where architecture content must be tailored to address a specific purpose that has been articulated, documented and is well understood.[8] They also require a certain amount of trust—that the data is valid and current, that the mode of collection is correct, that the vision expressed truly represents the desired output or outcome of the requirements, and that the execution of the solution will meet requirement expectations. However, adherence to a life cycle approach provides a level of rigor to the architecture process that eases these fears considerably.

Conclusion

The establishment of an architecture life cycle that mirrors the critical models of the NCOW Reference Model provides an additional level of security and support to the requirements process as it evolves within the DoD. The life cycle establishes that the data collected in response to a requirement is the critical element, the architecture views that are created represent that data to varying audiences so that these audiences can understand what is being stated in response to the requirement, and that this data must be actively managed to ensure that it remains current and useful.

Endnotes:

[1] Zachman, John, A Framework for Information Systems Architecture. *IBM Business Systems Journal 26:3 987*and *Extending and Formalizing the Framework for Information Systems Architecture. IBM Business Systems Journal 31:3 1992*

[2] The ASD (C3I) is now known as the office of the Assistant Secretary of Defense (Networks & Information Integration) (NII). Intelligence and Policies function previously under the ASD (C3I) were transferred to other offices within the Department.

[3] Information Technology Management Reform Act (Clinger-Cohen Act of 1996) PL 104-106, Section 5125, 110 Stat. 684 (1996).

[4] Net-Centric Operations and Warfare Reference Model (NCOW RM), v1.1, 17 November 2005

[5] DoD CIO/ASD (NII) Architecture & Interoperability Directorate

[6] Forward Compatibility for DoDAF is that Architectures developed in accordance with previous version can be recast using DoDAF 2.0. Concepts added in DoDAF 2.0 may not be able to be re-cast in DoDAF 1.5 or 1.0.

[7] CJCS Instruction 3170.01F, Joint Capabilities Integration and Development System (JCIDS), 1 May 2007

[8] DoD CIO/ASD (NII) Architecture & Interoperability Directorate