Beyond Web Services: Towards On-Demand Complex Adaptive Environments

Dr. R. Cherinka, Dr. R. Miller, and C. Smith
The MITRE Corporation
903 Gateway Blvd., Hampton, VA 23666
Phone: 757-896-8555, Fax: 757-826-8316
rdc@mitre.org, drbob@mitre.org, csmith@mitre.org

Keywords: Web Services, Semantic Web, Workflow Orchestration, Choreography, Complex Adaptive Systems, Policy-Based Computing, Quality of Service

Abstract

Complex adaptive systems are dynamically assembled systems characterized by multiple competing stakeholders, fluid requirements, emergent behavior, and susceptibility to external pressures that can cause change across the entire system. Net centric operations for the US Department of Defense (DoD) can be considered a complex adaptive system, representing a shift from traditional system-based interactions toward information-based web service transactions requiring highly secure, reliable, and dynamic "on-demand" capabilities. This net centric environment must accommodate unpredictable external factors that demand rapid response and flexibility to change. This paper presents an approach to modernizing toward global net centric operations for the DoD. We discuss key principles of complex systems engineering to consider, approaches to on-demand data and IT infrastructure strategy based on web service and semantic web technologies, as well as guidance and initial observations on spiral development and management using Communities of Interest combined with a Developer's environment.

1. Introduction

Complex adaptive systems are characterized as having unpredictable behavior, fluid requirements, multiple competing stakeholders, and are susceptible to external pressures that can cause change across the entire system. In many ways, thousands of loosely-coupled transactions across the web, choreographed in synchronous and asynchronous ways to represent dynamic and highly complex business models can be considered a complex system.

The US Department of Defense (DoD) net centric environment is a good example of such a system, with many unpredictable external factors that often demand rapid response and flexibility to change. Net centric operations for the DoD represents a shift from traditional system-based interactions toward information-based web transactions, adding the requirement for highly secure, reliable, and dynamic "on-demand" capabilities.

XML and web services are key technologies providing a foundation for this net centric vision. However, in order for an on-demand DoD to be realized, an evolution toward intelligent information exchange based on semantic web technologies as well as enhanced policy and resource management is required. This implies an evolution of the enterprise data strategy and IT infrastructure to support it.

This paper presents an approach to modernizing toward global net centric operations that MITRE is helping the DoD to adopt. We discuss key guiding principles of complex systems engineering to consider, insight into a desired on-demand data and IT infrastructure strategy based on web service and semantic web technologies, as well as guidance and initial observations on how to spirally develop and manage a complex system using Communities of Interest combined with a Developer's environment.

2. Background

Digital information rapidly is becoming integrated into all aspects of military activities. There is a goal across the DoD to find new and better ways of managing information and providing capabilities in response to quickly changing needs. The DoD has a large number of legacy and emerging systems that are making great strides toward achieving that goal. They fall short, however, in a number of areas. Most of them are still large, monolithic systems, each of which has to provide a full information management infrastructure (transport, network, data, interface layers,..etc). Because of this, there is only limited horizontal exchange of data amongst the systems--hence interoperability is a real problem. The systems are very configuration intensive and difficult to administer.

Furthermore, they are not very tailorable to a given operational environment. Finally, these systems have a very costly life cycle. Once fielded, keeping these products up to speed with the state of the art requires very costly upgrades, and replacement outright becomes cost-prohibitive.

One such DoD system is the Air and Space Operations Center (AOC), which is used as an example to form the basis for describing the engineering approach, challenges and observations discussed throughout this paper. The AOC today is assembled from over 80 elements. There are infrastructure elements, communication elements, applications, servers, and databases. The goal is to compose the desired capabilities from the elements found in, or which can be brought into, the AOC. For the most part, today's systems are not composable. The systems:

- Do not share a common conceptual basis.
- Are not built for the same purpose, or for use within specific (AOC) work flows, or for use exclusively at AOCs,
- Share an acquisition environment which pushes them to be "stand alone" ,
- Have no common control or management,
- Do not share common funding which can be directed to "problems" as required,
- Have many "customers;" the AOC is only one,
- Evolve at different rates (as do individual system components) subject to different (generally uncoordinated) pressures and needs.

Because of the above, integrating the AOC is an unbounded, unpredictable engineering activity. The AOC is thought of as a complex adaptive system, and as such there is a need to go beyond traditional systems engineering approaches [10, 12].

3. Emerging concepts on complex systems

Complex Systems are constantly changing. They respond and interact with their environments – each causing impact on (and inspiring change in) the other, usually through bottoms-up affairs, not top-down designs. Change ripples through complex systems causing local "pressures" among juxtaposed systems causing those systems to respond by undergoing change themselves. This is typically referred to as co-evolution, and in this way complex systems evolve - very much like what is seen within ecosystems. Some interesting characteristics of complex systems include:

- Dynamically assembled: often integrated from existing components
- Evolving requirements: typically articulated as vision statements or broad architectures.
- Emergent functionality/behavior: from the interaction of the components themselves w/o specific direction
- Crosses program boundaries: competition for resources & alternative solutions

Previous research has been accomplished to show that traditional systems engineering approaches do not work well when applied to complex adaptive systems [3, 4]. Instead, the notion of complex systems engineering has matured over the past few years as a way to address DoD enterprise engineering. Some key principles of this approach include:

- More emphasis on capabilities, less emphasis on requirements
- Focus on early discovery and evolution of composite behavior, functionality, and performance. This usually emerges upon integration and through the use of early prototypes
- Emphasize design guidelines, such as the use of layered architecture and open standards
- Use of rapid development spirals and experimentation

Throughout the remainder of this paper, we discuss an overall approach to enterprise engineering DoD complex systems commonly referred to as Net Centric Operations.

4. Net Centric operations for the DoD

Net-Centric Operations entails the networking of information producers (e.g., sensors), decision makers, and consumers to achieve shared awareness, increased speed and quality of decision making, and a higher tempo of dynamic operations [1, 2]. This concept of net-centricity motivates the following set of Enterprise Capabilities:

- Connectivity of users, applications and systems to shared, enterprise-wide services and information.
- Shared semantics and understanding of information across the enterprise.
- Unity of effort through distributed, collaborative operations and workflows.
- Predictable end-to-end performance across the enterprise.
- End-to-end secure enterprise operations.

The Net-Centric Checklist shown in Figure 1 depicts the DoD's overall strategy for achieving net-centricity across several categories: data, services, security and transport [13]. This checklist provides a basis for modernizing DoD systems and is based on several DoD and Industry best practices:

- Design application and system functionality as accessible and reusable services
- Expose service functionality through programmatic interfaces
- Maintain an abstraction layer between service interfaces and service implementations
- Describe service interfaces using standard metadata
- Advertise and discover services using standard service registries
- Communicate with services using standard protocols

Data
- Make data visible
- Make data accessible
- Make data understandable
- Make data trustable
- Make data interoperable
- Provide data management
- Be Responsive to User Needs

Services
- Service Oriented Architecture
- Open Architecture
- Scalability
- Availability
- Accommodate Heterogeneity
- Decentralized Ops and Management
- Enterprise Service Management

IA/Security
- IA Posture and Continuity of Operations
- Identify Mgt, Authentication, Privileges
- Mediate Security Assertions
- Cross Security Domains Exchange
- Encryption and High Assurance IP Interoperability (HAIPE)
- Employment of Wireless Technologies

Transport
- IPv6
- Packed Switched Infrastructure
- Layering and Modularity
- Transport Goal
- Network Connectivity
- Concurrent Transport of Info Flows
- Differentiated Mgt of QoS
- Inter Network Connectivity
- Joint Technical Architecture
- RF Acquisition
- Joint Net-centric capabilities
- Ops and Mgt of Transport Services

Figure 1. Net-Centric checklist

As the DoD community migrates toward Net-Centric operations, a set of criteria has been developed for use in measuring the maturity of progress toward achieving enterprise-level capabilities as a set of four levels. The first level pertains to systems adapting to the Net Centric environment in very basic fashion (e.g., basic network access (HTTP and IPv6) and data sharing). The second level pertains to collaborative information sharing across systems in the enterprise (e.g., all systems networked, web and information services). The third level pertains to intelligent information exchange across systems using semantically enriched publish, subscribe, query and brokering capabilities (e.g., self configuring networks, multiple security levels and combined coalition networks, data translation and mediation, service level agreements). Finally, the fourth level pertains to seamless operational awareness across systems in an on-demand distributed computing environment (e.g., dynamic bandwidth optimization, seamless cross domain access, seamless data interoperability).

5. Enterprise integration of complex systems

In this section we discuss a general 3-tier architecture to transition DoD programs into a highly reliable, distributed service based enterprise to meet on-demand net-centric requirements. Fundamental to this transition is the evolution of the enterprise-level information technology infrastructure, supported by key web service and semantic web technologies [6]. This evolving architecture leverage existing and emerging trends and capabilities in industry and will build upon previous iterations to enable greater degrees of information and service sharing across the entire enterprise.

5.1. Service-oriented architecture (SOA)

One of the most well established and widely accepted infrastructures used in industry today is the SOA. Through the use of web and information services, as well as open standards for information description, a SOA significantly improves the ability to share information and processes across an enterprise. Also, because SOA builds on technologies and standards that are greatly mature, COTS and open source solutions are widely available, and these solutions can be leveraged to reduce the work required in transitioning to a SOA.

The most important aspect of a SOA is the transitioning of individual capabilities across the enterprise into web or information services. By making capabilities available in this way, they are more easily accessible to a higher number of applications and users. Services also tend to be lightweight, allowing the enterprise to scale more effectively.

As services become available, federated registries allow them to be advertised in a reliable location. Users and applications can then browse or search the registry for services that meet their specific needs. In this way, overall situational awareness across the enterprise can be increased as capabilities that were once internal to an application or which were not well known can be found. This discovery can then lead to an evolution in business processes as well as increased effectiveness of applications and operators.

With capabilities transitioned to services, business processes can also be modeled as portable workflows, using emerging workflow open standards. Rather than creating applications that combine specific capabilities within a system, orchestration tools can be used to quickly select and arrange services to accomplish the same process. This flexibility results in a decrease in the number of stovepipe solutions required to support enterprise systems. Workflows also allow the enterprise to evolve faster and meet emerging needs more easily.

The use of a SOA also allows investments by other programs to be leveraged. Due to the loosely coupled and flexible nature of a SOA, individual components and capabilities can be provided by separate solution providers.

The infrastructure required to support SOA is composed of various tools and resources. These components are demonstrated in Figure 2.

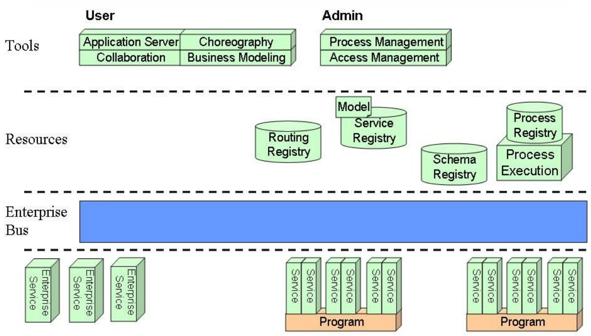

Figure 2. Service-Oriented Architecture

One of the most generic components supporting SOA is the application server. An application server is used to host and manage web-oriented applications and resources, including web and information services, to users within the community. Another generic infrastructure component is the set of collaboration services. These services allow participants within the community to collaborate with each other and are often provided as part of the application server.

Specific support for the service architecture begins with the schema and service registries. In the schema registry, the various standards for describing information and services are stored. The service registry is the next important component as it stores service advertisements. The use of service models in the registry increases service reuse and scalability by allowing similar services to be easily interchanged. When a service model is selected, a specific service is not selected until the service is actually used, so the routing registry provides a resource for storing service use per model.

Business modeling and process management are another large infrastructure requirement [5]. Support for this capability begins with the business modeling and choreography tools. Once the workflow is created, it can be stored in the process registry and optionally advertised as a service to allow for reuse. Business process execution is then carried out by an application resource, and this application can be monitored and managed by user and administrative tools.

5.2. Intelligent information exchange

Effectively managing information and its distribution is a difficult task within an enterprise. As the availability of information increases, an infrastructure supporting IIE is necessary to manage and maintain it. IIE enables this management through the use of ontologies and publish/subscribe/query interaction to locate and transfer information objects in an optimized fashion. However, information management across the enterprise is a difficult problem, and while there are some COTS solutions, the most useful capabilities are still being researched. IIE is the second phase of infrastructure modernization because it leverages SOA investments and because development of the outstanding features is well underway.

The use of information objects is key to IIE because it tightly binds information with the metadata that describes it. By sending information as an object, the recipient remains aware of the context long after the object is received; this decreases the potential for misuse or misinterpretation and increases potential reuse. As a result, the amount of redundant information exchange through the network is decreased and correct application of data is improved. The use of information objects can be integrated into a SOA by modeling objects using XML.

Ontologies are another important aspect of IIE because they increase semantics and provide a common representation for data throughout the enterprise. This ensures that participants within the enterprise can understand it. By using ontologies, a common and meaningful terminology is established throughout the enterprise, and both users and applications can reason based on it. This means clients will be able to put information to greater use as well as be able to collaborate more effectively. Investments in the SOA can be leveraged when adopting ontologies by using related open standards such as Resource Description Framework (RDF) and Web Ontology Language (OWL).

Sharing of information in an IIE system leverages the standard representation of information by providing a standard model for publication and subscription. In this model, information services have a common interface of disseminating information while consumers have a common interface for receiving information. Additionally, a standard query language ensures queries can be reused across multiple services. As a result, seamless exchange of applications with either publisher or subscriber roles can be accomplished and network resources can be better optimized.

All of these concepts finally combine to support information management in the form of an information brokering system. With a broker, information can be advertised, discovered and transformed. In this way, information producers can be sure that access to that information is significantly increased while clients can be sure they always have access to the most current information.

The infrastructure required to enable IIE builds on the SOA by extending the abilities of existing resource and applications while also providing new capabilities. The components necessary for IIE can be seen in Figure 3.

Many components already existing in SOA are leveraged in IIE. For some components, this means extended functionality, as is the case with the application server, collaboration capabilities, and access management tools. These components continue to provide the same functionally in IIE as in SOA, but they are updated to support those processes and capabilities which are specific to IIE. Additionally, other components can continue to be used without modification, such as the process creation, management, and execution infrastructure as well as the schema registry. For these components, no modifications are necessary because publishers and consumers act as services while information objects and advertisements use standards-based representations.

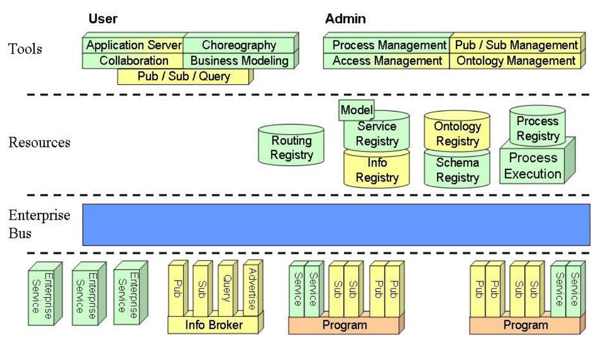

Figure 3. Intelligent Information Exchange Infrastucture

Beyond these existing components, there are many new components specific to supporting IIE. First of these components are those that support the use of ontologies. One of these components, the ontology registry, functions similarly to the schema registry by providing a standard storage mechanism for ontologies. Ontologies can be added, removed, and managed through the associated management tools.

Along side the ontology components are those components which are required to support the publication of and subscription to information. These tools allow users to publish, subscribe to, or query for information using a simple interface while allowing applications to perform the same interactions programmatically. Administrative tools then allow the resulting information routes to be managed.

Finally, the core services of the SOA are extended to include the information broker services. As a result, the service interfaces required for the broker can be made highly accessible and optimized.

5.3. On-demand computing (ODC)

As services and information become widely available throughout the enterprise, it becomes increasingly important to optimize the use of network resources. The most important aspect of this optimization is the leveraging of services and capabilities to share work across providers. To perform this automatic farming, however, requires a very complex infrastructure cable of dynamically shifting work between similar services and routing the results. While industry is starting to address these capabilities at the web service level, even partial support in COTS products is vastly immature. Since the maturity of the important aspects of on-demand computing is so low, these capabilities are planned as the long term architecture for C2 programs.

The use of unified resource models for advertising all types of resources in an on-demand infrastructure is core to the architecture's success. Through this system, information and services can be easily categorized and compared, allowing like or redundant resources to be identified and leveraged. Models also allow clients to select the type of resource required rather than a specific resource, thereby enabling reuse of workflows in different locations and as resources change in availability. The portability of these models relies on the use of open standard for their specification.

The primary benefit of this modeling system is then leveraged by a dynamic resource management system. By allowing clients to select types of resources, the dynamic routing system can then select a specific resource at the time of use. In this way, resource selection can be determined based on the current environment to achieve an equal split of workload between providers and to quickly handle loss of a service. This optimization is important in an operational environment where the redundancy across programs needs to be leveraged as an asset to provide increased response time and fail safe operation.

This resource selection, as well as all other decisions occurring in the enterprise, can be governed by the policy infrastructure [7, 11]. Providing policy specification and enforcement as an infrastructure service allows for uniform enforcement throughout the enterprise and eliminates the likelihood of redundant and conflicting policies. This uniform application of policy is important for ensuring commander's intent is enforced across the enterprise and elevated operator trust in automated functionality. Policy specification is another area where open standards would be employed to enable portability and common representation.

The primary goal of on-demand computing is to enable a highly optimized and reliable service and messaging infrastructure. This is important as it will ensure overall performance in the diverse network environment within a given C2 node and between it and its partners. As the need to collaborate and share information increases, an infrastructure made to do so in a timely, safe fashion will be important to establishing trust for a highly distributed community. An infrastructure supporting on-demand features will provide the reliable messaging necessary to build this trust.

The infrastructure required to enable on-demand computing builds on IIE and the SOA by extending the abilities of existing resource and applications while also providing new capabilities. The components necessary for on-demand computing can be seen in Figure 4.

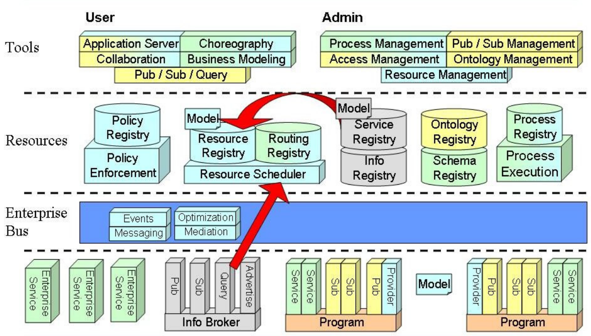

Figure 4. On-demand Computing Infrastructure

The on-demand environment builds on previous systems and components. As was the case in the IIE infrastructure, these common components require little modification or extension but are vital to proper operation.

The first large change required to support on-demand computing is support for a unified resource model. This requires a standardized handling of both information and services, so the once separate components required for each are merged into a resource registry. Process creation and execution also have to evolve to support the handling of resource models, thereby requiring changes to resources such as the routing and process registries and to tools such as the business modeler and process manager. Finally, since the information broker plays an important role in scheduling the exchange of information, its capabilities are transitioned into a resource scheduler capable of handling both information and services.

Policy management and enforcement also requires substantial additional and modification to the infrastructure. The first part of this addition is the inclusion of a policy registry and enforcement engine leverage open standards for policy description. The next part of this modification requires the access management tools to leverage the new policy specification. Policy can apply to many areas of computing, including access restriction, and a common interface is required to create and managed all policy.

The final addition needed for on-demand computing is support for reliable messaging. These core services augment the enterprise bus by providing standard handlers for transmitting messages in the network. In this fashion, events can easily propagate throughout the entire enterprise while guaranteeing delivery.

6. Developing complex systems

Net Centric Operations as a complex system has an effect on the DoD acquisition process, and to adequately address development and integration of complex systems, there is a shift of emphasis from building one-of-a-kind solutions to putting in place an environment and set of processes to help in the development and maturation of capabilities as they transform from innovation to fielded capability. In this section, we discuss two strategies being used across the DoD: Communities of Interest and the use of Developer's networks or environments [8].

6.1. Communities of interest (COI)

Interoperability, the ability to effectively share information and services, continues to be a difficult problem, both in the DoD and commercial endeavors. In addition, achieving a high level of interoperability is fundamental to realizing fully the benefits of SOA, IIE, and ODC. In today's complex environments organizations will communicate; build systems, services and interfaces; and transport, describe, and structure data in diverse ways. Interoperability requires that information producers and consumers come to terms with their vocabularies and manage their data so that both the producer and consumer have the same understanding of what the information means and how it is used. Interoperability also requires that these same producers and consumers define, manage, and register the service specifications to meet the requirements of on-demand net-centric architecture.

Attempts at data and vocabulary management typically lean towards data standardization; that is require organizations and services to implement the same data definitions and knowledge representations (vocabulary). Over the years, the DoD has invested heavily in common vocabularies with some successes [9]. But the goal of being able to share information widely remains elusive due to such factors as differences in culture and business practices. There is a large cost in designing and maintaining standardized data structures at the enterprise level. Further, it is increasingly difficult and costly for DoD systems to keep up with the pace of change in implementing these large vocabularies.

More recently the DoD is fostering vocabulary agreement on a smaller scale through a Net-Centric Data Strategy (NCDS) designed to support Service Oriented Architectures [14]. This strategy is designed to support the information exchanges found in loosely-coupled, complex system environments. The NCDS seeks to make all sharable data visible, accessible, understandable and interoperable by capturing and registering the associated metadata and posting all data to shared spaces to provide access to all users except when constrained by security, policy, or regulations.

In the net-centric environment, users and applications discover, post, and access information through both core and domain-specific services (SOA). The core services will be provided through Net-Centric Enterprise Services (NCES). Enterprise services include discovery, messaging, mediation, and collaboration services. Domain-specific services will build upon these core services to provide the mission capabilities needed to support net-centric operations. As specified in the DoD Net-Centric Checklist, that prescribes a framework of "design tenets" to assist program managers and organizations in becoming net-centric, net-centric services must be built on open standards (e.g., WSDL), be scalable, discoverable, accommodate heterogeneity, and support decentralized operations and management.

DoD COIs consist of information providers and consumers who must share information in pursuit of shared goals, missions, or business processes. COIs are similar to communities of practice with in the commercial sector. Some COIs may be large functional or cross-functional groups, while others will be smaller more expedient groups focusing on some more localized mission need or process. Regardless of their size, COIs will consist of information producers and consumers, as well as system developers whose role is to implement the NCDS and specify those services required for COI participants to interoperate.

Individual COIs will provide the necessary service specifications through open standards such as WSDL. They will develop the COI vocabulary and document the metadata via knowledge representations such as RDF and OWL. They will be responsible for registering their data definitions and metadata in the appropriate registries so they can be discovered to support data sharing. COIs will also be responsible for registering their service descriptions in services registries to support service discovery and usage. Consequently, COIs provide the mechanisms to employ the net-centric infrastructure and achieve IIE and ODC.

6.2. The developer's environment

Across the DoD several developer's networks are being matured as a way to address complex systems challenges. The intent is to create an environment where researchers, developers, testers, and users can meet and exchange their ideas, code and expertise as they experiment and productize new capabilities. The focus is on creating an environment and process (rather than a product) that facilitates 3rd party participation, eases entry and exit into the baseline of a system and minimizes integration "touch time" to achieve interoperable and integrated (loosely coupled) capabilities.

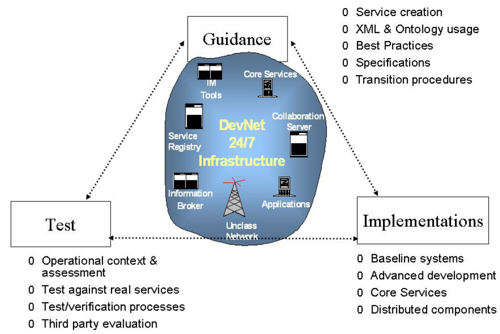

Figure 5. Characteristics of a developer's environment.

Figure 5 depicts the typical characteristics of such an environment, presenting both physical and collaborative aspects. Physical aspects include 24/7 infrastructure, applications, and services to support experimentation, initial concept development, and advance maturation (get your product to the next level). The collaborative aspects provide support to Communities of Interest through access to guidance, implementations, and testing information

The environment helps to integrate the user and developer through knowledge sharing, providing a process of evaluation, a mechanism of reward, common understanding of safety constraints, as well as rules for cooperation and competition. Typical uses include:

- Providing access to existing systems
- Providing various development levels of infrastructure, applications, and services
- Providing core services and infrastructure (e.g., service registries, brokering technologies, security) to enable rapid deployment, discovery, and usage
- Publishing guidelines for information service creation and usage based on accepted industry and government standards
- Enabling user and provider discussion and feedback channels for collaboration (e.g. forums)
- Ensuring usage and testing in operational context

Finally, this environment supports the collaborative documentation and understanding of requirements (e.g., certification) and procedures for transitioning services into production spirals for a system.

7. Observations

Based on our experience with the AOC, the following observations are made about what seems to work well from adopting the approach discussed in this paper:

- Architectural frameworks, vision documents, architecture products (UML), and technical roadmaps help manage and engineer the AOC as a mega-system
- Continuous involvement from COI members, and gaining consensus around infrastructure and tenets
- Active involvement of senior leadership and representative organizations
- Use of open standards, common vocabularies, capturing metadata
- Spiral development and Experimentation
- Developer's Network , integration facilities and environments (virtual and real)

Likewise, the following observations are made about what does not work so well:

- Difficulty in capturing requirements, especially in trying to describe how parts will work in context of the whole
- Implementing a common strategy across multiple stakeholders and getting everyone on a convergent path. Stakeholders need better guidance and criteria on implementing web standards & technologies
- Managing expectations and dealing with uncertainty (managing risk) across COI members, users, and senior leadership
- It still takes too long to the field capabilities, resulting in constant technology, expectation and user changes
- There is a lack of availability of core utility and mediation services
- Outdated Security policies which still serve need to hide vs. need to know

8. Conclusions

In this paper we presented an approach to modernizing toward global net centric operations that MITRE is helping the DoD to adopt. We discussed key guiding principles of complex systems engineering to consider, and how these principles apply to the DoD net-centric strategy. We highlighted an architecture and data strategy to support this, evolving from a service-oriented architecture to intelligent information sharing to on-demand computing. We then discussed the role of communities of interest and developer's environments are playing in this transition, and provided some key observations as to what is working and what is not thus far based on our AOC example.

Migrating to Net Centric operations will demand an unprecedented degree of cooperation and coordination among all stakeholders. Efforts will be started at different times in different places but will all need to be brought into line. While Web services standards and technologies enable interoperability, they do not guarantee it. Complex systems theory and extensive experience demonstrate that sufficiently complex systems need evolutionary engineering strategies.

9. References

[1] D. Albert, J. Garstka and F. Stein, Network Centric Warfare, CCRP, 1999.

[2] D. Alberts, J. Garstka, R. Hayes and D. Signori, Understanding Information Age Warfare, CCRP, 2001.

[3] Y. Bar-Yam, Dynamics of Complex Systems, Perseus, 1997.

[4] Y. Bar-Yam, When Systems Engineering Fails---Toward Complex Systems Engineering, 2003 IEEE International Conference on Systems, Man & Cybernetics, October 5–8 Washington, D.C., USA, 2003.

[5] J. Cardoso, Quality of Service and Semantic Workflow Composition, PhD University of Georgia, Department of Computer Science, Athens, Georgia, 2002,

[6] R. Cherinka and R. Miller, Lessons Learned in Using Web Services for Enterprise Integration of Complex Systems in the Department of Defense, International Conference on Systematics, Cybernetics and Informatics (SCI), July 2004.

[7] R. Cherinka, C. Smith, C. Wild et. Al., Towards Policy-based Information Management for the Joint Battle Space InfoSphere, IEEE Policy 2003 Conference Proceedings, 2003.

[8] R. Cherinka, R. Miller and S. Matechik, Air and Space Operations Center Developer's Network, MITRE Next Centric Experimentation Conference, August 2004.

[9] R. Miller, M. A. Malloy, E. Masek, and C. Wild, Towards and Information Management Framework, Information Knowledge Systems Management Report, 2001

[10] J. Moffat, Complexity Theory and Network Centric Warfare, CCRP, 2003.

[11] C. Nicodemos and A. Damianou, A Policy Framework for Management of Distributed Systems, PhD Thesis Imperial College of Science, Technology and Medicine, University of London, Department of Computing, London, 2002

[12] D. Norman and M. Kuras, Engineering Complex Systems, MITRE Technical Report, 2004.

[13] OSD/NII Net-Centric Checklist, Version 2.1.3, 12 May 2004

[14] OSD/NII Net-Centric Data Strategy, 9 May 2004