10TH INTERNATIONAL COMMAND AND CONTROL RESEARCH AND TECHNOLOGY SYMPOSIUM
The Future of C2

Title of Paper:
Agile Assessment Techniques for Evaluating Mission Capability Portfolio Ensembles in Complex Adaptive Architectures

Topic: C4ISR Architecture

Author: Jack Lenahan
POC: Jack Lenahan
Organization: Office of the Chief Engineer
Space and NAVAL Warfare Systems Command
Charleston, S.C.
Address: P.O. Box 190022
N. Charleston, South Carolina: 29419
Phone: 843-218-6080
Email: John.Lenahan@Navy.mil

Abstract
Given the prototypical architectural template's demise, the purpose of this research is to begin a formulation of the "Agile Assessment Methodology" needed to evaluate the mission capability impact of using composeable web services in complex adaptive architectures. Network Centric Warfare (NCW) Assessment Processes must validate that a "rush towards a transformation" by date "X" does not sacrifice warfighter capability by introducing de-stabilizing architecture components. What assessment methodology and criteria will be used to evaluate or even define the NCW architectural boundaries for platform system reductions? What assessment methodology will be used to evaluate mission execution success probabilities given the migration away from traditional platform centric mission capabilities? The results of this research indicate that platforms should be wary of removing systems in favor of GIG services which may jeopardize crew or platform survivability; it also recommends that composeable assessment and simulation capabilities required to manage the assessment of mixed architecture capability ensembles be developed.

Introduction
The advent of Network Centric Warfare (NCW) architectural paradigms which will replace legacy systems is introducing new challenges to traditional assessment methodologies. In order to maintain availability, Web Services (singular or composed into capability packages), will need to exist in an environment composed of highly available GRID architectures, with agent based availability monitors. But what is the mission impact of such architectural novelties? What process can be used to answer the following questions?

1. What is the impact of the migration away from platform centric capabilities?
    a. What systems stay on a platform?
    b. What systems can be removed from a platform?
    c. What can be removed and replaced by a Service Oriented Architecture (SOA) pub/sub capability?
2. How will mission capability portfolios be impacted by the appearance of GRID Architectures overlain with Agent and Service Oriented Architectures?
3. How will "on demand" self service assessments be conducted for composeable web services?
4. How will asset allocation and asset management policy agents be assessed?

Network Centric Warfare (NCW) introduces several categories (see figure 1 below) of transformation which should be assessed both separately and together. The categories of transformation are the migration towards a Service Oriented Architecture (SOA) and the associated reduction of platform centric software for mission support, composeable capability ensembles derived from the web services resident in the SOA, and the notion of sharable assets. Let us discuss platform footprint reduction assessments first.

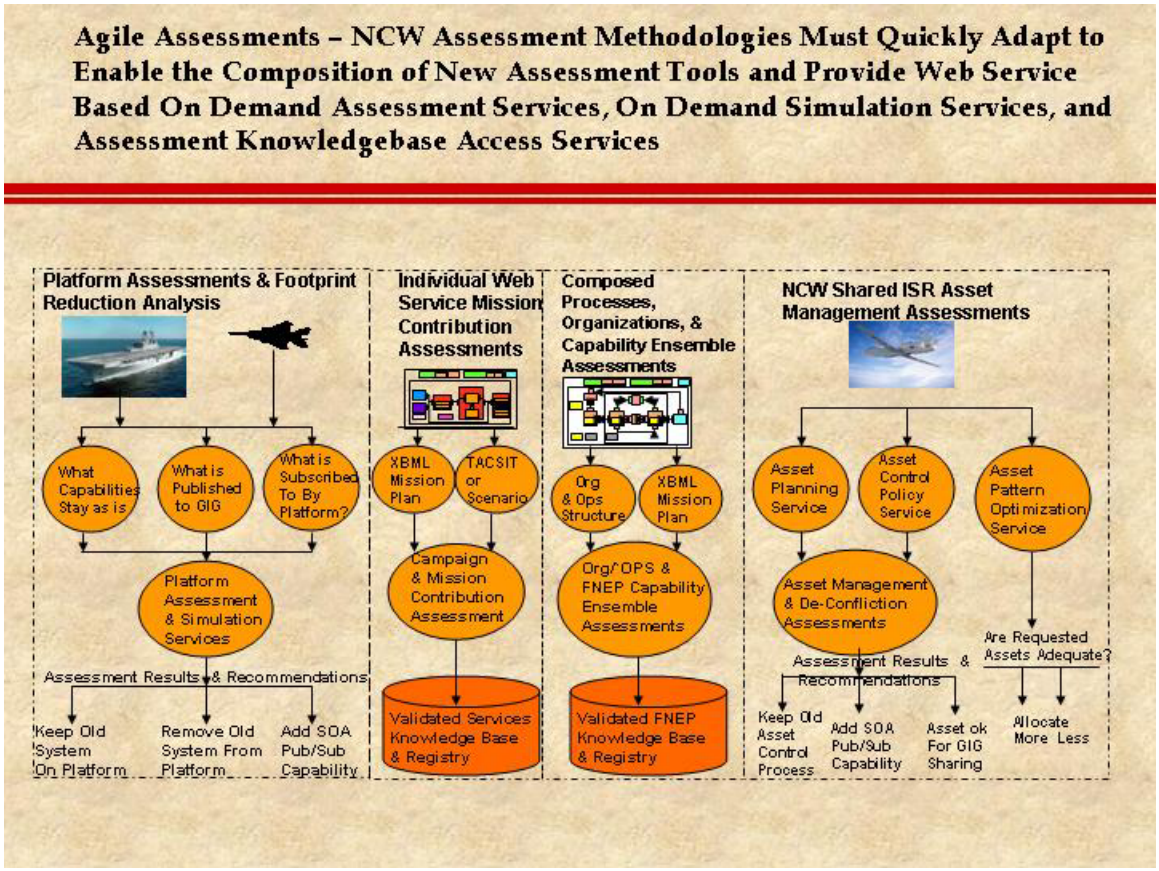

Figure 1 – Depiction of Assessment Types Requiring Composeable Assessment Services, Web Tools and Processes Agile Enough to Adapt to Changing Conditions

Assessing the Platform Impact of NCW: Evaluating the advantages of Platform Centric Application Software versus Platform Subscriptions to GIG Web Services

De-Platformization - what stays and what goes?

Very little NCW guidance has been provided in terms of what capabilities "make sense" to remain on a platform. One of the major types of assessments which must be performed in the early phases of NCW analysis is deciding what platform based capabilities can be replaced by web or GIG services and what should remain on the platform. There are several assessment issues for this type (what stays on a platform) of assessment. First, in the case of an IA attack, simple network failures, lack of redundant communications or general disconnectedness, can the platform and crew survive? In other words, is a platform's survivability increased or decreased by a movement towards non-platform based web services? Can the platform still achieve mission success? Second, is the risk associated with De-Platformization worth the benefit of the data which might be published? In other words, does anyone else care about the data that would be published from platform maintenance computers and is anyone likely to care about such data? Third, can the current platform capability actually be equivalently replaced by a GIG service? Fourth, is the cost to remove the current platform system exorbitant?
The figures below illustrate the issue. This may seem like a trivial and possibly trite example, but it illustrates my point rather well and simply. The graphic below depicts a current architectural model of a particular auto pilot. All required components needed to provide the auto pilot capabilities are on board and probably contain some redundancies built-in.

Figure 2 – Autopilot Architecture[1] with all system components on board in one unit.

The following figure depicts the opposite of the above graphic. Namely that only the sensors and the actuators remain on board (by definition they must), the data processing needed to generate the proper information for the actuators has been removed and replaced by GIG based web services. Thus, the web services – somewhere on the GIG – now are mission critical and must be totally responsible for the health of the auto pilot. New risks have now been introduced in terms of Communications Availability, Service Availability, GIG Availability, and Data Availability. Information Assurance (IA) is now a potential new risk also.

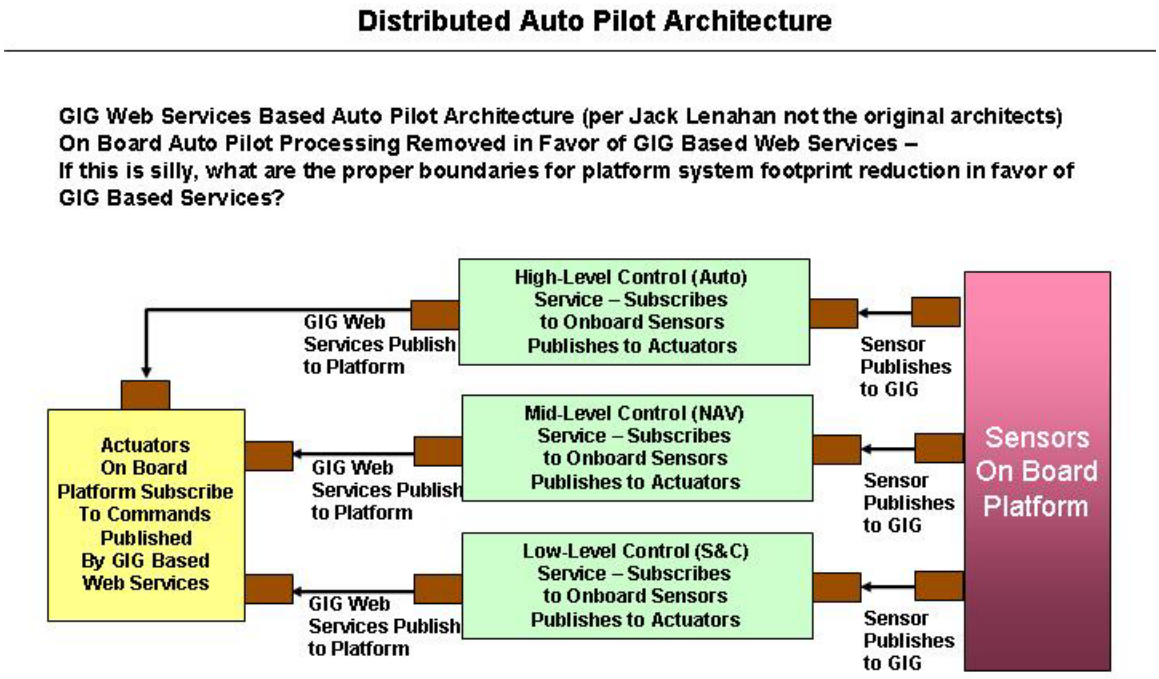

Figure 3 – GIG Based Web Services Auto Pilot Hypothetical Model

What should one conclude about these obvious differences? What is the proper boundary of Net Centric Distributed Services, Platform Survivability, and simple common sense? What are the criteria that architects should use to consider platform system footprint reduction? This issue for obvious reasons has led to many heated and frustrating debates. This author believes that in the absence of clear and simple OASD/OFT guidance, that at a minimum, platform computational services needed for day to day platform operations and survivability should not be in the realm of candidates to be removed from platforms for footprint reduction purposes or some theoretically pure SOA implementation. "Platform Survivability" at a minimum should form a stringent boundary of acceptable systems removal.

Assessing Service Oriented Architectures and Composed Service Ensembles

In a companion paper[2] also prepared for this conference, I state that a standalone SOA will be insufficient in terms of providing infrastructure stability. I proposed that a highly available, disaster recoverable, GRID model (overlain with availability and performance monitoring agents) be implemented in order to sufficiently cover the reliability, performance, and availability issues needed for combat missions. A short discussion of several key points from that paper follows below.

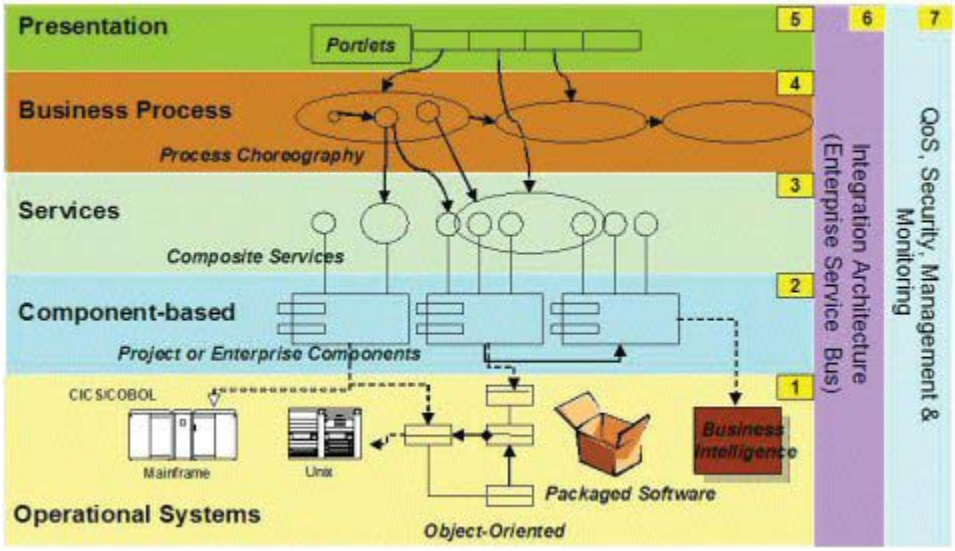

Figure 4 – Relationship of an SOA to other software architecture stack layers

Discussion of the SOA model

The model[3] above probably depicts the early implementations of NCW software architecture. The layers provided by the article's authors show that:

Layer 1 is the current legacy software architectures and suites of applications in various languages and configurations, (the source for data and potential services). This layer is not redundant, may or may not be stateful, and has no DR capability. Thus, if a failure occurs at this level, no sophisticated restart or graceful failover is possible. DoD must decide if the initial versions of NCW will invest in the cloning of older legacy systems for HA/DR requirements satisfaction or if they are willing to accept the risk of failure which may impact a missions success.

Layer 2 is introduced as the componentization of some subset or all of the legacy system's data and capability. This is not the SOA layer. This is an attempt at standardization of functionality in a more modular fashion such that the service or data provided by the components (old legacy system capabilities) can be accessed by a wider

audience in a standard manner. Note that this graphic does not depict any vehicle to support HA/DR or scalability.

Layer 3 is the first formal SOA layer. This is where the services and data either exposed through component interfaces, or written as new web services are resident, registered, users authenticated, and made available for search engines. As the authors state, the services can exist as individual expositions or as composite web services. Again, by itself this SOA layer of UDDI, Single Sign On, Content and Service Management services are un-managed from a reliability perspective. No HA/DR or scalability exists.

Layer 4 is the business process layer of the SOA. It is at this level that both orchestration and / or choreography occur. Note that this is usually supported by commercial orchestration products such as BPEL. BPEL by itself is not HA/DR or easily scaled. Commercial products also introduce license and unique security issues which may make single sign on difficult.

Layer 5 is the presentation layer. This layer is (as the author's indicate) deliberately decoupled from the SOA below it. But once again this causes issues for the SOA infrastructure architect. This is yet another layer for single sign on and for HA/DR design. How is this to be handled in the event of user surge or failures of the portal stack?

Layer 6 is the Enterprise Service Bus. This is a good method for hybrid integration, but it once again introduces single sign on complications, HA/DR and performance issues. Thus the ESB (if the SOA is implemented this manner) must be HA/DR.

Level 7 is the first level at which monitors are introduced and they are also the first peek at HA/DR, and performance management. The figure below depicts the software architecture which would be overlain on a GRID as Highly Available.

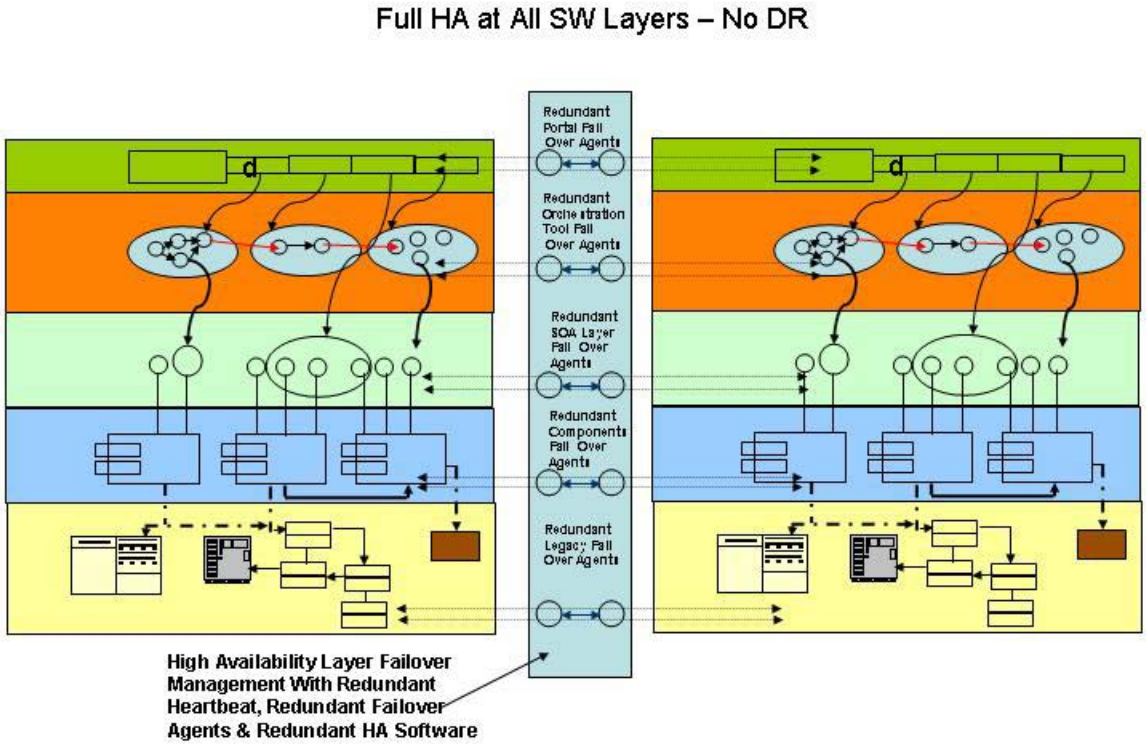

Figure 5 – Highly Available SOA Stack

The HA depicted is obviously a clone of the standalone layer graphic but with heartbeat and failover management software.
The next graphic depicts a HA/DR but with the entire HA layer sets cloned and located at a geographically removed location from the HA layers The graphic also introduces the need for DR failover monitoring agents which are also duplicated for the other main HA requirement of no single point of failure.

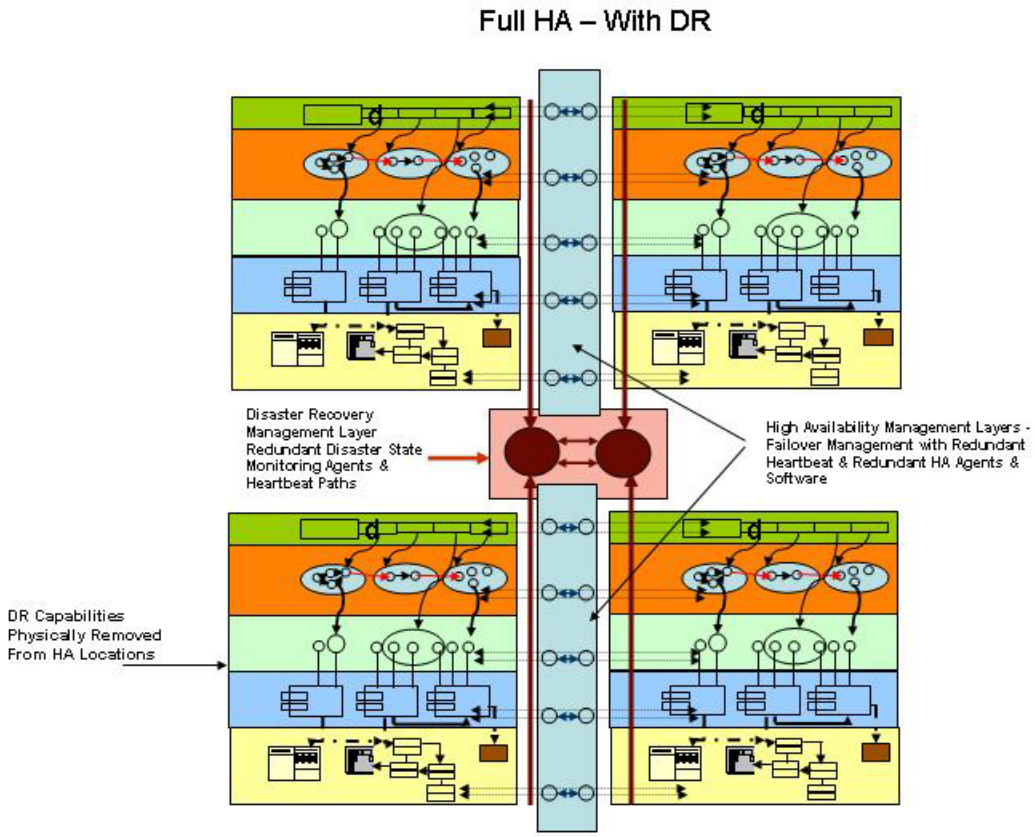

Figure 6 – Highly Available SOA Stack with DR Software Architecture at a Geographically Different Location than the HA Software Stack

Commercial industry stops at this level of redundancy as the standard definition of five nines, thus, so will the author accept this depth of backup as a boundary. Now that we have covered the HA and the DR, we still have one more requirement set to cover, performance due to surges in users or workload due to a new conflict. The only commercial model currently available to even attempt dynamic routing of service executables due to performance is the architecture known as a GRID. The GRID architecture depicted below would be overlain with the software layers depicted in the above figure. The GRID is designed to manage both HA, DR, and re-allocate software due to poor Service Level Agreement or Quality of Service behavior (SLA/QoS). If a service is not meeting the required SLA, GRID performance agents can move the software to a freer node in the GRID. Indeed, the GRID scheduler probably will schedule services for execution using computation resources with the lightest workloads in the first place in order to preemptively manage future computational or other resource bottlenecks. The formal definition[4] that I am using is as follows:
A GRID is an "IT infrastructure that (1) supports dynamic resource allocation in accordance with service-level agreement policies, efficient sharing and reuse of IT infrastructure at high utilization levels, and distributed security from edge of network to

application and data servers and (2) delivers consistent response times and high levels of availability—which in turn drives a need for end-to-end performance monitoring and real-time reconfiguration."

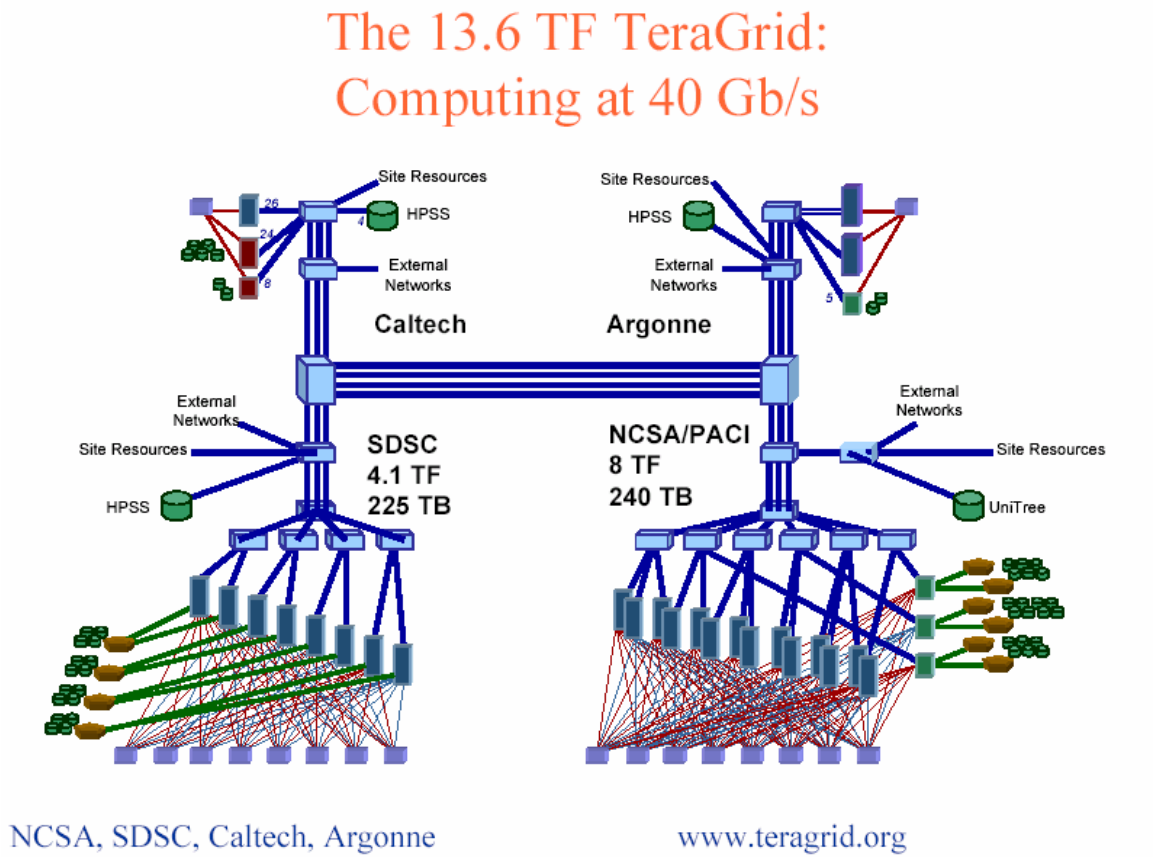

Figure 7 – GRID Physical Architecture – A Complex Adaptive System which is scaleable, highly available, disaster recoverable, and capable of dynamic program execution resource re-assignment.

Please note that the above diagram[5] depicts the paper's intentions with respect to HA and DR. Thus, in this diagram, the left side contains redundant physical networks, computers, and storage, if the networks, storage and computers each contain failover and heartbeat monitors, then the left side is totally infrastructure HA. By adding or overlaying the HA software layers redundantly, then the left side is SOA and Infrastructure HA. The fact that this is a GRID permits dynamic resource reallocation and scheduling if a particular web service begins to experience poor or degraded performance. Thus the architecture "adapts" to failure as well as degrading performance. Performance monitoring agents can have degradation thresholds set at any arbitrary level to immediately attempt to re-allocate the degrading services' executables to a more adequate resource set upon threshold crossing detection. The existence of the right side infrastructure at a physically different location (CALTECH vs. Argonne) qualifies this instantiation as disaster recoverable if we also overlay the additional DR agents and layers.
The fact that this architectural model can adjust to individual component failures as well as poor performance makes it easily fit into the definition of a Complex Adaptive

Architecture. Note: For a detailed presentation of these concepts, please review my paper "Are Service Oriented Architectures the Only Valid Architectural Approach for the Transformation to Network Centric Warfare?" available of the conference CD or website.

Discussion of the Lenahan QoS scale

How can we rate whether or not a given proposed architectural solution is adequate in terms of reliability, availability, and performance? The answer is through the use of the Lenahan Quality of Service Scale for NCW C4ISR Architectures. The table below describes the levels of reliability that I am recommending be meta tagged at the web service description level or at a capability ensemble level.

QoS Level	Capability or Capability Sets exposed as Web Services	State Recording	Simple State Recording with graceful fail over management by simple agents	Agent Monitoring of All Web Services in given C4ISR Architectural Orchestration or Choreography for Graceful Recovery of Services (Also applies to each service and its orchestration tool in a given FNEP being fully Stateful and agent monitored	HA (All enabling software / hardware infrastructure layers (Listeners, Authentication SW, Firewalls, Single Sign-on Software, Directory and Naming Management, MOMS, Database Software, Redundant Directories, Redundant data, SAN, NIC, etc) for the entire orchestration set)	HA with Full DR - Clone Of HA Suites	HA/DR with guaranteed performance management (GRIDS with all 7 ISO Layers HA/DR)
1	Y	N	N	N	N	N	N
2	Y	Y	N	N	N	N	N
3	Y	Y	Y	N	N	N	N
4	Y	Y	Y	Y	N	N	N
5	Y	Y	Y	Y	Y	N	N
6	Y	Y	Y	Y	Y	Y	N
7	Y	Y	Y	Y	Y	Y	Y

Table 1 - Lenahan Levels of NCW Reliability, Availability, & Performance QoS

If web services contain their own meta tags defining how they comply with the above reliability, availability, and scalability levels, then SLA\QoS expectations can be managed. But the tag must describe the entire set of software and enabling software infrastructure as to it's HA/DR rating not just the web services. Also, if a particular web service rating is too low; funding decisions can be made using this rating scale as a justification.

Discussion of each Quality of Service compliance level for reliability, availability, and performance

1 Web Services are implemented as new software or exposed existing legacy capability. No state recorded, No failover, HA, DR at this level, just the web services themselves are available, registered, discoverable, etc.

2 Web Services add State Recording (at least start or finish states with possible intermediate states recorded to a database table or a state recording service). No failover, no HA, no DR at this level. However, this positions the service for upward QoS scale movement. I realize that this is contrary to many current definitions of web services. I am recommending that we move in this direction for the achievement of greater availability, stability, and disaster recovery ability.

3 Web Service Simple State recording with graceful fail over management by simple agents. Please note that the remainder of the architecture or physical infrastructure may not include state recording, just the web services at this level. Also note that qualification at this level would not necessarily include all the web services in a given orchestration or choreography. No HA, no DR.

4 Agent Monitoring of All Web Services in given C4ISR Architectural Orchestration or Choreography for Graceful Recovery of Services (Also applies to each service and its orchestration tool in a given FNEP being fully Stateful and agent monitored). Thus, single sign-on to all services is HA, the orchestration and choreography tools are HA, and the web service monitoring agents and the web services themselves are all HA at this level). But the enabling software stacks, the portals, Apache listeners, and the enabling infrastructure do not need to be HA.

5 Complete architectural stack is Highly Available - The web services, orchestration and choreography tools, all enabling software / hardware infrastructure layers (Listeners, Authentication SW, MOMS software, Firewalls, Single Sign-on Software, Directory and Naming Management, Database Software, Redundant Directories, SAN, NIC, etc, for the entire orchestration set) – all hardware, networks, OS, and communications are automatically failed over. This means that if a legacy system is the source of the any of the services in the orchestration sequence and the legacy system is not fully HA, then the orchestration sequence defaults to a QoS level of 4. In simple terms every possible piece of software and hardware in all services in a given orchestration or choreography sequence are HA.

6 Full Disaster Recovery and first 5 nines level but without guaranteed performance management by dynamic re-allocation of compute resources or storage resources. This is level 5 plus a highly available DR clone.

7 Full Disaster Recovery and second 5 nines level but with guaranteed performance management by dynamic re-allocation of compute resources or storage resources. All layers and the orchestration / choreography tools, single sign-on, and the orchestrated web services

and their source computer sets fully HA/DR. Performance is not impacting availability.

An agile assessment methodology for NCW should now include "dynamic" or "near real time" assessment of the QoS levels for any composed or pre-composed mission capability set. Services used individually and as composed sequences (orchestrated or choreographed or both) or composite engagement packs such as the NAVY's FNEPs[6], (FORCENET Engagement Packs are a set of pre-composed web services orchestrated and choreographed for a particular set of mission capabilities), should be granted a Lenahan QoS rating so that the user understands what QoS level to expect. For example, if a given orchestration sequence of 10 serially called web services is executing at QoS level 5 and the Orchestration tool fails at sequence number 4, then the remaining web services will never be called or activated if the orchestration tool itself is not manually restarted, gracefully failed-over, or disaster failed-over or GRID state-managed into transparent automated graceful fail over and restart. But if the sequence is executing at Lenahan level 7, then it will be automatically restarted and the remainder of the sequence will be called.

Assessing composed service ensembles

Per the discussion above, the infrastructure needed to reliably sustain web services or web service ensembles should be Highly Available, Disaster Recoverable, and Scalable. A GRID architecture with a Lenahan QoS rating of 7 will provide this level of reliability. But what about the assessment of the actual composed package itself. How should we approach this subject?

First, if we were to only assess individual services, then we will not understand their behavior when orchestrated or choreographed or called by "N" users. So assessing individual services, while necessary in itself, reveals little about the "Set of Services" required to successfully execute all the tasks needed to support missions from start to finish.

Second, individual service descriptions may be inadequate to support rapid composition into "mission ensembles of orchestrated capabilities". This is part of the so called ontology and semantics problems associated with "sense making" of web service knowledge. The following example may be useful. Suppose someone codes and registers a web service which has only the following description: "tank re-supply service". What does the description mean? Is this an Army Tank munitions replenishment service, a KC130 gas tank re-supply service, a fighter jet gas tank mid air re-fueling service provided by the KC130, or an Army tank's gas tank refilling service from a fuel farms gas tank? Without sufficient descriptions, the subscriber must actually investigate the service (assess the particular service at the content level) to determine if the web service meets the needs of the user. Attempting semantic resolution at the "last moment" before a service is required may be too late. Instead, suppose that we decide that we know what the top 50 missions of the JOINT services are and that we attempt to compose a "Pack of Services" in advance, and then tag the pack itself for UDDI discovery but at a great level of META tag detail so that the tags are semantically rich enough to support the mission types required, then we have just

described the justification and meaning of the notion of a FORCENET Engagement Pack.

Discussion of FNEP Concepts

FORCENET Engagement Packs are small scale system ensembles which demonstrate the engagement enabling power of FORCENET (the NAVAL version of NCW) by integrating Joint sensors, platforms, weapons, warriors, networks, and command & control systems with requirements to perform cross-mission enabled Network-Centric Combat Reach Capabilities. In Network Centric Terminology, the systems will be replaced by the SOA layers of composed web services. Let's evaluate a hypothetical FNEP based upon the graphic below, keeping in mind that the actual services composing these packs will be resident at various locations on the GIG and also will exist in random states of data content accuracy and timeliness.

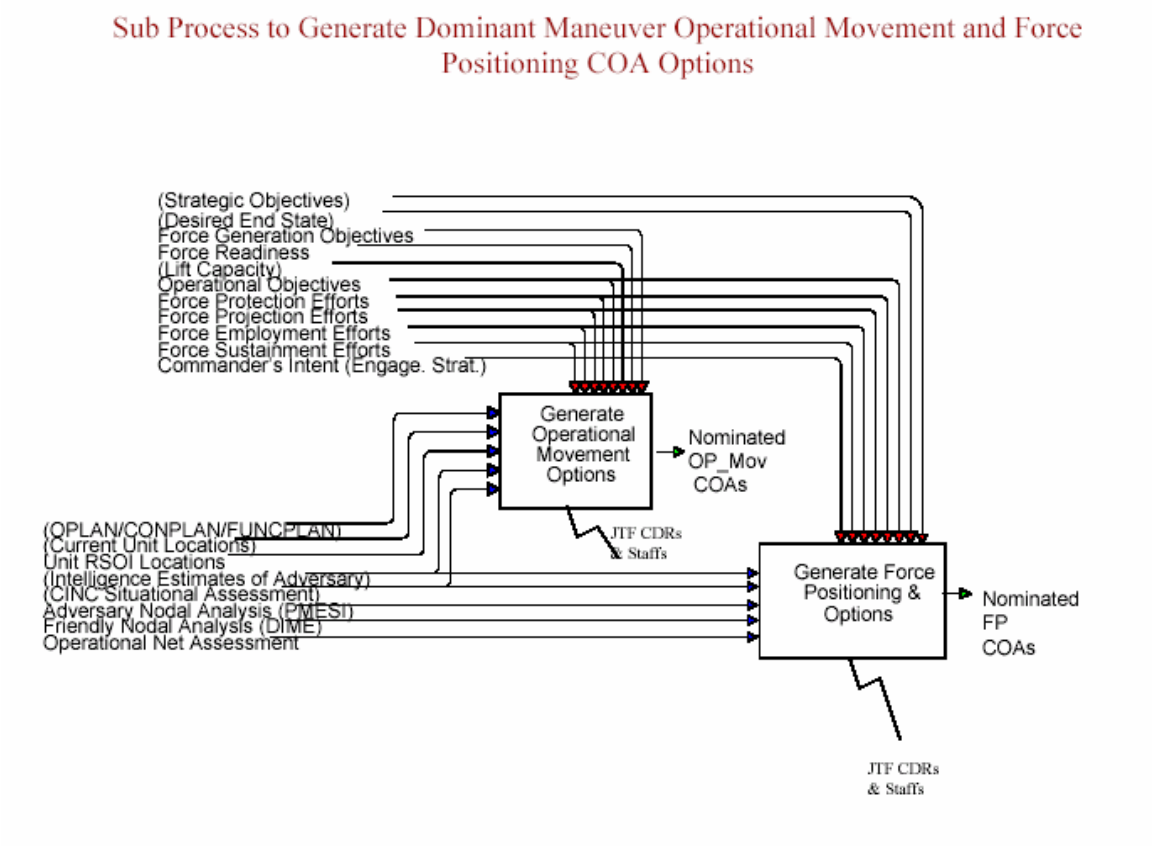

Figure 8[7] – Force Positioning Creation Process Using SOA Based Web Services

The following is a summary analysis of this web service model of generating and publishing force positioning (FP) courses of action. The graphic above depicts the IDEF model of NCW for FP creation. In this hypothetical example, the process must "publish" a Force Positioning Set of Alternatives (Courses of Action) containing the sections mandated by the process controls. Through the following analysis, I hope to make a single and extremely important point: That the composition of "new NCW

capability" through the compilation and use of UDDI registered and discovered services, is much more complex and difficult than the NCW PowerPoint engineering rangers would have folks believe. It is this author's opinion, that it is extremely risky to attempt to compose new capabilities on the fly through the vehicle of a single architect "sitting in front of an orchestration tool, armed with his UDDI search results". The following analysis, I believe, should help to demonstrate that "Engagement Packs" or "Capability Ensembles" composed by both user defined "Assembly Agents" and teams of operations experts and human architects, with their product (the engagement pack) itself registered as a "discoverable service", is a much more prudent methodology. The table below itemizes the list of mandatory web services required to publish force positioning courses of action. A causal glance at the table reveals several key points: first, that many services will be required, second that the services will need to be available with current data content (timeliness attribute of NCW), third, that the approvers will need subscription and publication capability services, and fourth that there must be either a "content aggregation service or a set of human capable interactive services in order to produce and publish the required sections of the force positioning plan.

Controls Services	Input Services	Approvers Services	Output Services
Desired End State	**OPLAN/CONPLAN /FUNCPLAN**	**Army/MC FP COA approver**	**Strategic Objectives**
Force Generation Objectives	**Current unit locations**	**AF/NAVY FP COA**	**Force Sustainment Efforts**
Force Lift Capacity	**Unit RSOI Locations**	**SPEC OPS U.S. COA Approver**	**Force Lift Capacity**
Operational Objectives	**Intelligence Estimates of Adversary**	**SPEC OPS U.K. COA Approver**	**Desired End State**
Force Protection Efforts	**CINC Situational Assessment**	**SPEC OPS Poland COA Approver**	**Force Generation Objectives**
Force Projection Efforts	**Adversary Nodal Analysis**	**SPEC OPS Spain COA Approver**	**Force Employment Efforts**
Force Sustainment Efforts	**Operational Net Assessment**	**CJTF Sec Def**	**Force Protection Efforts**

Table 2 – Web Services required for the Force Positioning Web Service to be Published

In the decomposed process model below, I have depicted the IDEF model as a set of nodes; each node will perform a different task required in the production of the FP COAs. However, in my hypothetical process, I have added mandatory approval levels, in this case, the approvers are JOINT and Coalition level commanders (therefore, each approver, in "N" languages" must see the finished product appear in a portlet for his/her approval, thus an Approvers Web Service & Workflow Service must exist to accomplish this besides the actual service of creating the FP), who must agree with the selection of the Course of Action. Each task node depicted below also shows that a subscription service with the capability to digest multiple publications' data contents must exist.

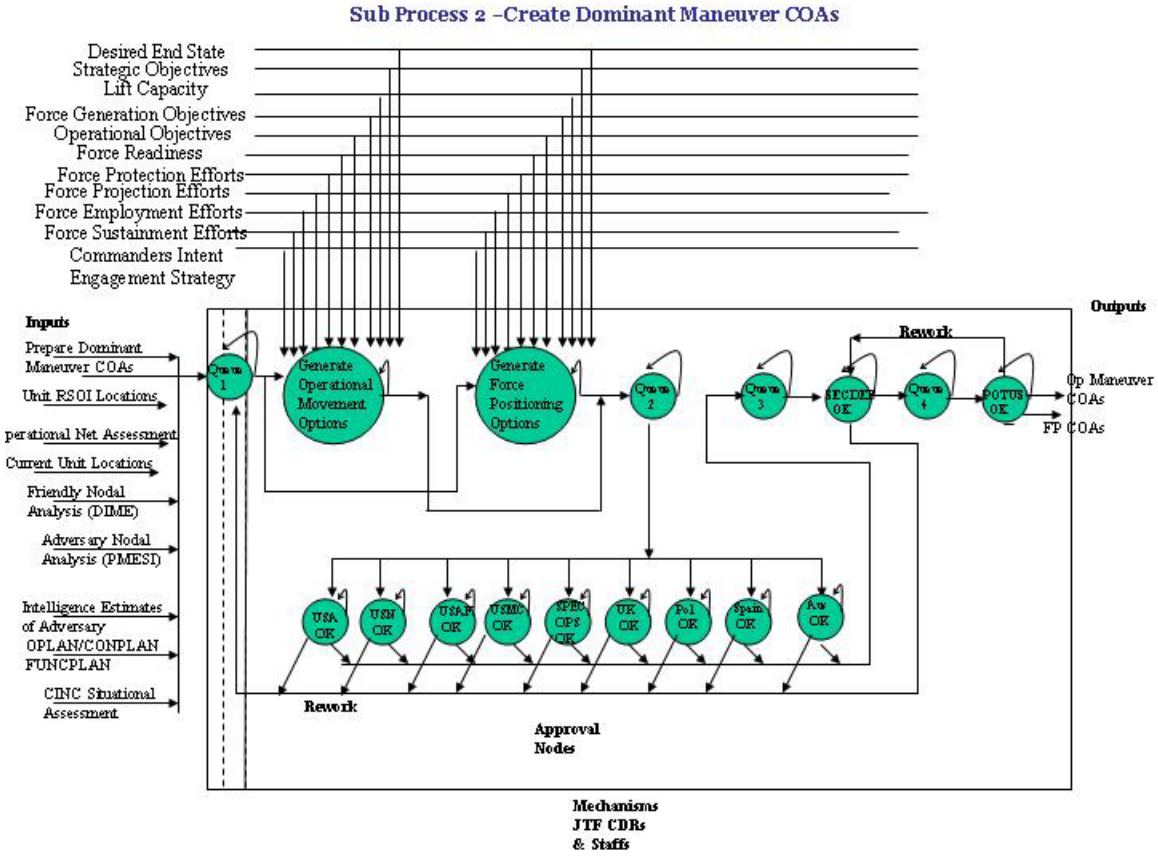

Figure 9 – Web Services required for a Force Positioning Publication Service in an for SOA based model

Now let's determine what the service pack must consist of and what assessment services must exist in order to properly evaluate and rate the "Engagement Pack". The following "controls level" web services must exist: Strategic Objectives, Desired End State, Force Generation Objectives, Force Lift Capacity, Operational Objectives, Force Protection Efforts, Force Projection Efforts, Force Employment Efforts, Force Sustainment Efforts, and Commander's intent engagement strategy. The following input services must exist: OPLAN/CONPLAN/FUNCPLAN, Current unit locations, Unit RSOI Locations, Intelligence Estimates of Adversary, CINC Situational Assessment, Adversary Nodal Analysis (PMESI), Friendly

Nodal Analysis (DIME), Operational Net Assessment. The following outputs must be generated and published individually and collectively as a FP plan: Strategic Objectives, Force Generation Objectives, Force Lift Capacity, Operational Objectives, Force Protection Efforts, Force Projection Efforts, Force Employment Efforts, Force Sustainment Efforts, Commanders intent engagement strategy, and the Desired End State. A total of 8 input services subscribed to, 10 controls web services subscribed to, and 10 output services published. How many subscribing services (to input and control publications) are there 1, 10, 18 a number in between? How many approval services are needed: One per language, one per JOINT service member, one per Coalition Partner Country? We can easily tally a minimum of 40 + separate web services (10 output, 8 input, 10 controls, 12 approver services, plus the workflow services (orchestration and choreography services) which will be required to perform the force positioning process. Some of the Assessment implications of this model in question form are:

1. What is the reliability/availability level (Lenahan QoS rating) of each service? The pack of services? Answer, the pack reliability is only as high as the weakest HA/DR/Scalable Service Component.
2. What is the probability that all the required services will be available when needed? Answer: this is both a probabilistic function and a resource de-confliction probability function (probability that you will get the drone whenever you need a fresh surveillance). Thus, the combined probability that a given service "x" will be available and current exactly when needed, and the probability that you will get drone control (or any asset control) exactly when you need it for your mission. I see asset de-confliction (where the source of data content is a sensor on a physical asset) as a major hurdle in the construction of so called sensor grids. Unless, Asset Scheduling Agents and associated web services are implemented via policy agents, I see little hope of peaceful asset allocation.
3. What resource scheduling policy agents will be used to perform de-confliction of scarce resources if re-flys by the drone are needed to confirm damage levels? Who will own these agents and define the asset allocation policies? How many more web services will this introduce is yet another assessment type.
4. What if all the services listed above are not web services but an unknown mixture of web services, open architecture CORBA registry services (object Oriented Method Remote access calls to applications), and legacy applications which still utilize MOMS or flat files for integration? It is my opinion that a switch will not be thrown on day "x" and then "poof" all legacy applications will disappear and be replaced by extremely stable and reliable web services. Therefore, I feel quite safe in stating that a "mixed architecture model" will be the core of NCW for quite some time into the future. This mixed architecture model will by all means require that a very robust and extremely agile Assessment toolkit be available as services on the GIG itself, to be able to manage the composition of mixed architecture Engagement Packs. If we have to find each of these by ourselves and understand the "composed behavior in advance," it may be quite a chore. However, suppose that the vendor or FN architects compose a capability ensemble in advance of operational usage, delivering force positioning as a

composite service, the vendor can then publish both the composed services in a UDDI for discovery and each individual service should also be placed in the UDDI for discovery by other potential users or FN architects or elegant ensemble composers for service portfolio management simplification. Thus, the following will need assessments in the above Force positioning model in a mixed architecture ensemble:

- Each service's Lenahan QoS rating for HA/DR and scalability expectations
- Gap analysis to determine if mandatory (mission critical) QoS gaps exist at the HA/DR scalability levels
- Timeliness probability of each service being current at the exact time it is needed
- Integration risks of mixed SOA and Legacy compositions.

Asset Allocation Assessments

In NCW theory, ISR assets will need to be managed as shared resources. It is assumed that this only means that the asset will be placed into some shared pool when its owner is not utilizing it. This is not true. A certain group of theater assets will be placed into a shared asset pool, the actual usage of the asset is to be determined by competing GIG policy agents. This new and revolutionary concept will require two types of assessments at a minimum: First asset allocation timeliness, second asset sufficiency assessment. The first assessment type is simple to define, mainly that the asset will be available when you need it. The second is much more complex. The asset request should be adequate to perform the mission. What if the composed mission is search and rescue and needs a certain amount of coverage by a drone aircraft in order to cover a certain search pattern thoroughly? The mission composer may not be technical enough to determine if he has requested enough drones to properly cover the search area, therefore an assessment of the proper number of assets should be dynamically performed upon completion of the initial mission formulation for the search. What if 6 drones are needed but only 4 are available? The assessment service for this asset class must be able to re-compute an optimal search pattern for the 4 drones assigned. Dynamically reconfiguring the drones' flight patterns in flight may be necessary also to support this type of activity.

How will "on demand" self service assessments be conducted for composeable web services or FNEPs? The introduction of Composeable Assessment and Composeable Simulations as a Possible methodology to provide Agile Assessment Capabilities for Assessing Mixed Architectures

What sort of tools can attempt to perform these assessment and simulation activities in the future? The complexity of FNEPs executing on a GRID is depicted below. The reason that I modified this NASA graphic is to demonstrate the complexity of FNEPs executing in a Highly Available, Disaster Recoverable, and Scaleable GRID or Complex Adaptive System environment. The FNEPs must be assessed by themselves, and then the

supporting infrastructure must be evaluated. The policy agents needed to permit asset allocation and management must also be assessed.

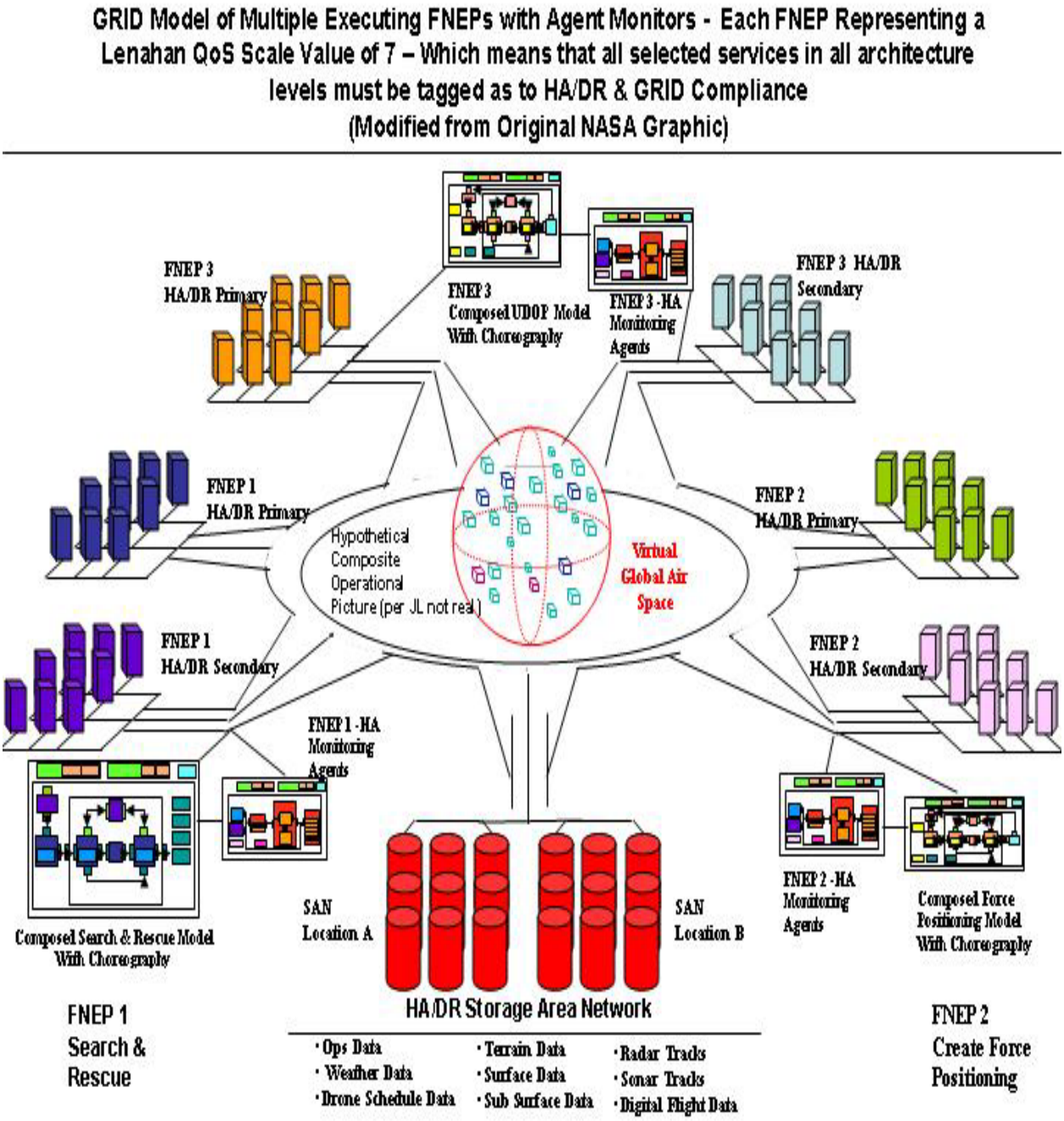

Figure 10 – GRID[8] model of multiple FNEPs to exemplify the Force Positioning Web Service Model (FNEP 2 in Graphic) as a FORCENET Engagement Pack – Note that the FNEP 2 package in this graphic itself has availability and scalability monitoring agents on the GRID and coexists with other executing FNEPs. In this case FNEP 2 is Lenahan Level 7 and FNEP 2

is also double UDDI registered as a discoverable service at the engagement pack level with each of the services composing the pack also registered on the GIG individually so that other engagement packs can be composed

What would an assessment and simulation architecture that could attempt to accomplish these assessment goals look like?

The graphic below is my suggestion to accomplish the composeable assessment and simulation capabilities required to manage the assessment of mixed architecture capability ensembles.

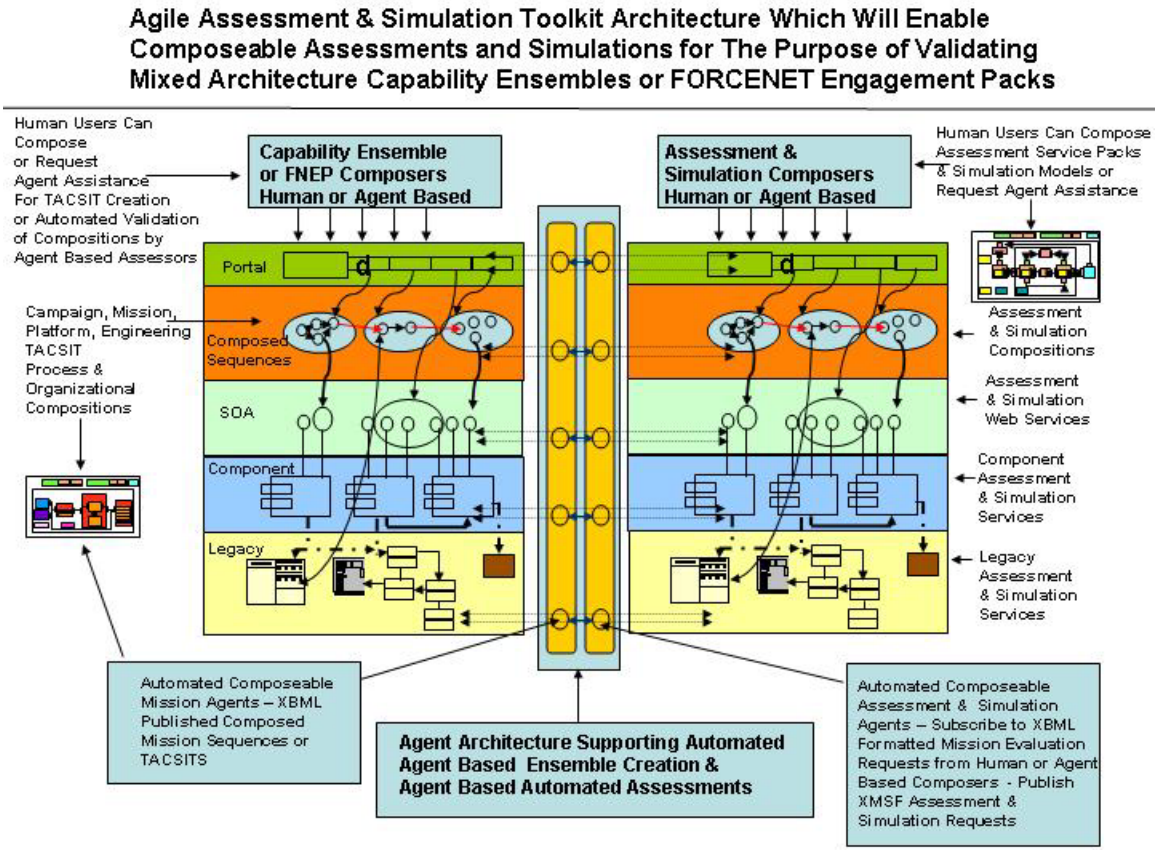

Figure 11 – Composeable Assessment and Simulation Toolkit Architecture to Support Agile Assessment

Starting with the mission composer, the composer would define the TACSIT or mission using the tools available on his portal. The user may select from a variety of web services, components, or legacy capabilities (This is the mixed architecture model that I believe will be the structure of early NCW implementations). The user must define all required assets and create a process or sequence of activities to be performed by the asset and services combinations. The user must also define the proper organization and chain of command structure which will govern the instantiation of the FNEP. The policy agents for sharing the assets will need to be interrogated for asset availability during the execution period of the FNEP. The

composer should be able to request assessments of his/her compositions at any time in the composition. The composer must also be aware of the reliability requirements that he/she is requesting. Thus, it is important that the Lenahan rating be computed automatically as an assessment service for the entire pack or segments of the ensemble. This is why I believe that composeable assessment and simulation services will be required to support ad hoc assessment requests. Standards will need to drive this solution or else little can be accomplished. There has been much research accomplished in the standards area which addresses my concerns. It is out of scope for this paper to provide a detailed discussion of the standards required but there are 3 which should be considered as a minimum set: C2IEDM, XBML, and XMSF.

Results

The NCW architectural constructs which will appear in the near future have not been subjected to formal rigorous engineering analysis. The underlying assumptions of NCW indicate that novel architectural formations, as yet defined, will appear as a major result of architectural composeability, particularly with respect to the use of so called "composeable web services". To quote Dr. Alberts: "No one …. can speak with final authority on NCW orthodoxy. NCW is, and will continue to be, the product of many fathers..."[9]

Besides the novel architectural constructs, the NCW architectural boundaries for platform system reductions have yet to be formalized. This author does not accept the notion that a radical reduction of platform system footprint can occur in the near (15 years) future. A total "pub/sub" platform design has yet to emerge in the NCW literature. Replacing platform centric mission critical systems (such as terrain avoidance radar processing, autopilot, flight stabilization, etc.) in favor of subscriptions to "stabilization services" published by compute agents somewhere on the GIG which have subscribed to pubs from the very platform that they are attempting to stabilize, is simply "mythological". OSD/OFT owe the services deeply considered guidance covering which aspects of on-board platform processing it considers to be "on the table" for removal from the platforms in favor of a pub/sub sequence.

In order to reduce risk, arbitrarily orchestrated or choreographed web service ensembles will require near real time, self service assessment of the composed "mission capability threads" hosted by complex adaptive architectures. This research concludes that a dynamic and composeable set of mission capability evaluation services, based upon XBML and XMSF can be used as the basis of an evolutionary and revolutionary capability. This author concurs with and embellishes other researcher's efforts; particularly the analysis[10] developed by Tolk, Hieb, et al. Composeable NCW "Self Service Assessment" services should follow the following guidance:

- **Develop Modeling and Simulation Web services that can be distributed via the Web**
- **Transform and tag existing data representations to international Joint standards (by using standardized Coalition data models such as the Command and Control Information Exchange Data Model (C2IEDM))**
- **Evaluate the applicability of XBML for Global Information Grid (GIG) Enterprise Services (GES) and Warfighter Services.**

- **Create Self Service Composeable Assessment Services which will provide Rigorous Pre-Mission Validation of the Composed Mission Capability Ensembles as hosted in Agent Monitored GRID Architectures**

References and Relevant Research

1. Autopilot Architecture - Source: Unmanned Dynamics, LLC - http://www.u-dynamics.com/sensor_fusion/sensor.htm
2. "Are SOA's the only valid Architectural approach for transformation to Network Centric Warfare?" – Lenahan, Jack 10th International Command and Control Research and Technology Symposium
3. Delving into Service-Oriented Architecture, By Bernhard Borges, Kerrie Holley and Ali Arsanjani - September 17, 2004 (Please note that I have both included the original and modified the original, the publisher of the cited material has requested that I post the following legal information: basically that no commercial implementations or sales of the material can be pursued without their permission. WWW.Developer.com/design/article.php/3409221 - **http://www.jupitermedia.com/corporate/legal.html - Jupiter Media Notice 1. Copyright, Licenses** MAY NOT MODIFY, COPY, REPRODUCE, REPUBLISH, UPLOAD, POST, TRANSMIT, OR DISTRIBUTE, IN ANY MANNER, THE MATERIAL ON THE SITE, INCLUDING TEXT, GRAPHICS, CODE AND/OR SOFTWARE. Subject to more specific terms on individual JUPM web sites, you may print and download portions of material from the different areas of the Site solely for your own non-commercial use provided that you agree not to change or delete any copyright or proprietary notices from the materials (certain areas require paid license fee prior to downloading any material). You agree to grant to JUPM a non-exclusive, royalty-free, worldwide, sub licensable, perpetual license, with the right to sub-license, to reproduce, distribute, transmit, create derivative works of, publicly display and publicly perform any materials and other information (including, without limitation, ideas contained therein for new or improved products and services) you submit to any public areas of the Site (such as bulletin boards, forums and newsgroups) or by e-mail to JUPM by all means and in any media now known or hereafter developed. You also grant to JUPM the right to use your name in connection with the submitted materials and other information as well as in connection with all advertising, marketing and promotional material related thereto. You agree that you shall have no recourse against JUPM for any alleged or actual infringement or misappropriation of any proprietary right in your communications to JUPM. **and Idea Submissions.** Domestic and International copyright and trademark laws protect the entire contents of the Site. The owners of the intellectual property, copyrights and trademarks are JUPM, its affiliates or other third party licensors.
4. "The Physiology of the Grid An Open Grid Services Architecture for Distributed Systems Integration", Ian Foster1,2 Carl Kesselman3 Jeffrey M. Nick4 Steven Tuecke1 1 Mathematics and Computer Science Division, Argonne

National Laboratory, Argonne, IL 60439 2 Department of Computer Science, University of Chicago, Chicago, IL 60637 3 Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292 4 IBM Corporation, Poughkeepsie, NY 12601

5. "Grids and High Performance Distributed Computing" - Andrew Chien , March 31, 2004, CSE225, Spring 2004 - The TeraGrid project is funded by the National Science Foundation and includes nine partners: NCSA, SDSC, Argonne, CACR, PSC, ORNL, Purdue, Indiana, and TACC. Any questions or comments please email the webmaster@teragrid.org.
6. FORCEnet Engagement Pack - *Technical Process Overview* - Phil Charles, Command Chief Engineer, Code 0E SPAWAR System Center Charleston FnEP Work Shop, 18-19 FEB 04
7. "An Abstract Process and Metrics Model for Evaluating Unified Command and Control - A Scenario and Technology Agnostic Approach ", Lenahan, Jack, Force Positioning IDEF model by Terry Mayfield of IDA
8. NASA's Information Power Grid - NASA's Virtual Air Space, William E. Johnston, Project Manager, Arsi Vaziri, Deputy Project Manager, Tom Hinke, employment Task Manager, Leigh Ann Tanner, Implementation Manager
9. "Network Centric Warfare - Current Status and Way Ahead", Alberts, David, Journal of Defence Science , Volume 8, Number 3, September 2003
10. **"** Developing Battle Management Language into a Web Service", Hieb, Tolk, Sudnikovich, Pullen