17th ICCRTS
"Operationalizing C2 Agility"

Composable Environments

A systems architecture for agile user-driven command and control

Topic(s)

Topic 3: Data, Information and Knowledge
Topic 7: Architectures, Technologies, and Tools
Topic 2: Approaches and Organizations

John Kruse, PhD and Todd Reily
The MITRE Corporation
202 Burlington Road
Bedford, MA, USA 01730-1420

Point of Contact
John Kruse, wkruse@mitre.org, 781.271.6069

Composable Environments

A systems architecture for agile user-driven command and control

John Kruse, PhD
The MITRE Corporation
Bedford, MA, USA
WKruse@MITRE.org

Todd Reily
The MITRE Corporation
Bedford, MA, USA
TReily@MITRE.org

***Abstract*—This paper outlines an ongoing investigation of how we might better construct an ecosystem in which data and computing resources are continually composed, decomposed and recomposed to create agile command and control applications that move data up the value chain so that they can become the knowledge that feeds decision-making and action.**

Ultimately, the desire for ever-greater agility will drive our C2 systems towards rapid composability in the hands of the end users rather than engineers. This shift of control and discretion to the operational edges will have sweeping implications for our organizations, processes and people. We discuss a supporting ecosystem of composable data and computing resources that can be developed to support the sharing of data and resources within requisite levels of security. If brought to fruition, such a flexible and agile C2 infrastructure could fuel paradigm shift in the acquisition, development and evolution of automated systems.

The recommendations in this paper are informed by extensive user observation, interviews, and personal experience working in operational environments. The scope of this work encompasses not only the operational users, but the acquisition, engineering, and governance communities that support operations.

Keywords: composable, mashup,

"Any intelligent fool can make things bigger and more complex...It takes a touch of genius and a lot of courage to move in the opposite direction" – Albert Einstein

I. Introduction

Every day, the experienced and qualified professionals that seek to supply command and control (C2) systems to warfighters try to do something that in the abstract seems simple – gather requirements, understand the requirements, decide on a plan, build it, and then field it. Nevertheless, we are beset by failures, disappointment, cost overruns, cancellations, and obsolescence.[1] Even when we do see deliveries of new C2 systems, they are often not quite right. Either the mission or environment has changed since the requirements were gathered, or maybe we just didn't quite nail the true essence of the warfighter need. We have reached a point where these failures seem to outweigh our successes and truly astonishing system breakthroughs in C2 systems just don't happen anymore. Meanwhile, the commercial world is experiencing a flood of innovation. How can this be when we are expending so many resources and employing so many talented people?

This paper will seek to first explore why it is so difficult to succeed in our current environment. Then we will lay out the plans for a new ecosystem of composable capabilities that may be further composed, extended and evolved to one day supersede our current systems.

II. Problems with our World

A. How We Arrived Here

To understand why things are the way they are, we first must look back at our history and how C2 systems came to be developed. Our information technology (IT) acquisition processes are a direct outgrowth of the period following WWII and the explosion in innovation in science and technology. Nuclear weapons, the jet and space ages, not to mention the constant tension of the cold war, all focused a tremendous amount of energy and resources on the development of evermore-complicated systems. Computers were a necessary link that could offer a level of responsiveness, predictability, and reliability that wasn't available previously to coordinate and control the many "working parts" of a complicated project or system.

A great example of the approach pioneered in this period is the race to the moon. After the wakeup calls of Sputnik and Yuri Gagarin, the United States made a manned visit to the moon a national priority, and President Kennedy had set a very tight timeline.[2] As such, the engineers and managers involved took the logical step of creating and enacting a painfully detailed plan of all the research, developments and decisions that would need to take place in order to put men on the moon. Although there were delays and unforeseen issues, the roadmap was executed successfully and in July of 1969, men walked on the moon.

The processes that are used today by the U.S. government and the associated players (e.g., contractors) to acquire complicated systems are much like those used in the Gemini and Apollo programs during the 1960s. We still attempt to create a detailed and exhaustive process with many checks and decision points along the way to ensure that the system works according to plan (e.g., DoD 5000). Over the years, more layers

The research discussed in this paper was funded by the MITRE Corporation's Innovation Program

of oversight have been added in a well-intentioned effort to avoid past mistakes. This has tended to render the system less agile and more cautious than might be necessary in the current environment.[3]

Nevertheless, there is some evidence that this approach does not work well for the development of information technologies (IT) as can be witnessed in reports from both the Defense Science Board and the National Academies on how to improve the acquisition of information technologies in the DoD.[3][4] Both cite the cumbersome and bureaucratic process and both recommend a new process specifically because the acquisition of IT is fundamentally different than the acquisition of weapons or hardware.

B. *Complexity, Dynamics and Intelligent Design*

There are two primary drivers that cause the current process to fail systematically with respect to information technologies – complexity and dynamics. In the case of complexity, we often see evidence of systems that have evolved over decades to reach high degrees of complication. Now, as we try to replace them, either in whole or in part, we find that they are extremely difficult to understand and/or decompose within the context of their use. This is complicated by the fact that often, the users want improvements on the current system, not just a one-for-one replacement. This heightened level of system complexity often leads to either analysis paralysis – in which we are unable to decide what to do – or deliveries that miss the mark and underwhelm the end users.

The second issue, the dynamics of the system's environment, implies that we don't have as much time to return with solutions as would be dictated by the current processes. Any solution that takes three years to complete will not achieve success if the requirements are changing at a significantly faster rate. In effect, this problem leads us to a situation where we are constantly solving yesterday's problems.

We believe that these two issues are very much akin to those faced in the commercial IT world that have really popularized agile development approaches. If a system is difficult to understand and/or it is changing rapidly, you have little choice but to quickly evolve solutions. In a nutshell, DoD processes take an "intelligent design" approach in assuming that if we can just gain enough information about the system and the environment, we can design and build great solutions.

If, however, the environment is either unknowable, and/or rapidly changing, this assumption no longer obtains. Instead, one must take an evolutionary approach, or as John Gall more artfully stated, "*A complex system that works is invariably found to have evolved from a simple system that worked. The inverse proposition also appears to be true: A complex system designed from scratch never works and cannot be made to work. You have to start over, beginning with a working simple system*."[5]

III. The Transformitive OODA Loop

We started the analysis for this work by looking at the current IT acquisition process; specifically at the core functions being performed. Simply put, the process is seeking to (1) gather requirements, (2) understand them, (3) create a plan for delivering a capability based on those requirements, and then (4) enact the plan. We realized that this could easily be mapped to Colonel John Boyd's model for decision and action commonly referred to as the "OODA Loop" for the acronym of the major steps – *Observe, Orient, Decide and Act*.

This model is widely used in the military and is popular in large part because it is both simple and provides insights into how to better operate in the world. The major tenet that was Boyd's foundational assertion is that in a competitive engagement "all parties go through repeated cycles of Observing, Orienting, Deciding and Acting, and whoever can go through the cycle consistently faster will win".[6] We felt that although the acquisition environment does not have an adversary per se, it is, nevertheless, subject to the same type of "evolving reality that is uncertain, everchanging, unpredictable"[sic] and would thus be a reasonable application of the OODA Loop.[7]

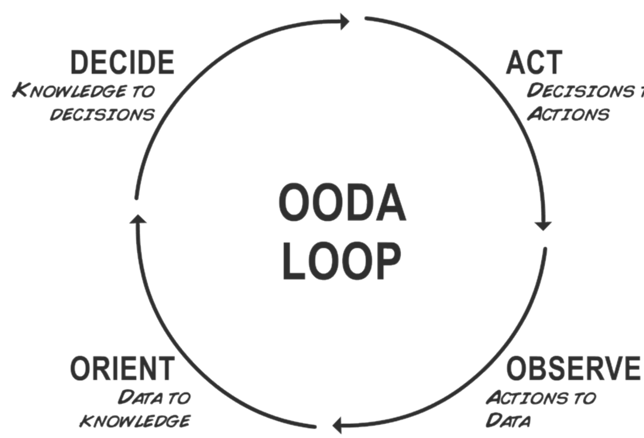

Figure 1, Boyd's OODA Loop

We also realized that each step in the OODA loop is a transformation of sorts. Observation is concerned with observing the real world which results in data, orientation takes that data and shapes it into knowledge that in turn fuels decisions which are then made real through action. Each step fuels the next as in Figure 1, Boyd's OODA Loop.

IV. Criticality of the Orient Phase

Our model is focused primarily on the creation and evolution of C2 capabilities and the facilitation of decision-making. As such, we are most concerned with the observe, orient and decide phases of the OODA Loop. These three phases are generally in the realm of the commander and the staff, while action is, for the most part, in the purview of subordinates actually executing the orders generated through C2 processes. Figure 2, C2 Opportunity rearranges the OODA

Loop in order to highlight the portion of the model that is largely controlled by the commander and staff.

It is important, however, to understand that each phase of the OODA loop is not created equal. Boyd recognized that the orient phase is the most critical and described it as “an interactive process of many-sided implicit cross-referencing projections, empathies, correlations and rejections.” Purposeful action simply cannot occur without the understanding afforded by orientation.

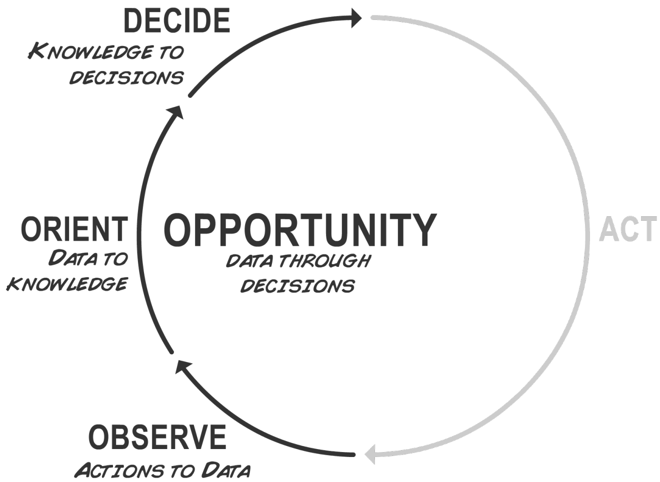

Figure 2, C2 Opportunity

Moreover, orientation is really at the core of command and control. In the military sphere, forces are not simply gaggles of individual warriors. They are concerted groups unified in their purpose and pursuit of common goals. Moreover, they have structured roles, hierarchies of command, and standardized procedures in order to facilitate coordination and lower barriers to working towards a common mission.

V. The Data to Knowledge Value Chain

Within the orient phase, we can begin to break down the specific tasks that the staff takes on in taking data and turning it into knowledge in order to fuel decisions from the commander. **Error! Reference source not found.Error! Reference source not found.**Figure 3, The Data to Knowledge Value Chain, illustrates our efforts to break up the orient phase and flesh out these staff tasks.

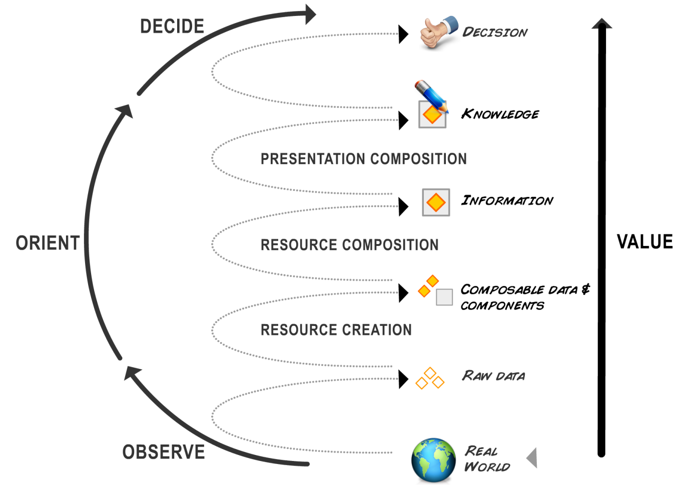

Figure 3, Data to Knowledge Value Chain

1) Resource Creation

At the lowest level in the observe phase, raw data is pulled from the real world and introduced into our value chain. Upon entry to the orient phase, this raw data must be transformed into a common format where it can be easily composed within the system. In some cases, this will be very simple and in others more difficult. For instance, an RSS feed would likely only require very simple reformatting, while a raw data link stream might require substantial translation or transformation. The key is that the common format acts as known and predictable “integration bowtie” that makes a world of data easily consumable and composable within our ecosystem.

2) Resource Composition

Once data is in a common format it can be transformed into information that is in a form that people can more readily understand. In our proposed ecosystem, this would be accomplished through the composition of data with computing resources or components (e.g., widgets). For example, a string of location data that depicts enemy positions may be of great interest, however, it is of limited value to the staff until it is combined with a map widget so that the data can be understood in context. By combining a data set with an appropriate display mechanism, the user is raising that data to the level of information and by doing so, raising its value to the organization.

We believe that data will be manipulated and transformed in other ways beyond display. Translation and transformation may also happen in this step. For instance, a target location expressed as Military Grid Reference System (MGRS) coordinates may be of limited value to an Air Force operations center because they tend to work with coordinates expressed as latitude/longitude couplets. Thus, the simple ability to easily translate between these two standards might elevate data to information for such a user in this particular context.

3) Presentation Composition

Information in turn is analyzed, synthesized and understood by the users who, in turn, create knowledge. For the purposes of our model, knowledge is communicated understanding. Any number of information sources may be required to create the understanding that is then communicated as knowledge. Understanding can only exist in the mind(s) of people in a given domain. Though that understanding is useful to the individual, it must be communicated to provide value to the greater organization. We term this task *presentation composition* because the major visible activity is the creation of presentations of the understanding (i.e., knowledge) that exists in the mind of the staff member(s).

VI. The Knowledge Chasm

The adoption of a model based on the OODA loop also helped us to better characterize behaviors which we had witnessed many times in operations centers and on military staffs – the use of Microsoft PowerPoint as a focal point for staff processes and decision making. As mentioned previously, staff members gather information, or create information themselves from data sources in order to increase their depth or breadth of understanding. Unfortunately, they are usually unable to continue working with the systems that bring them the information. As currently designed, *information* systems don't support the transition to knowledge. PowerPoint, however, flexibly allows users to paste in information and add their own assessments to create *knowledge* presentations. Thus, we repeatedly witness what we term a "knowledge chasm" in which the utility of information systems ends and users are forced to laboriously shift to a presentation system (e.g., PowerPoint) to continue up the value chain (Figure 4, The Knowledge Chasm).

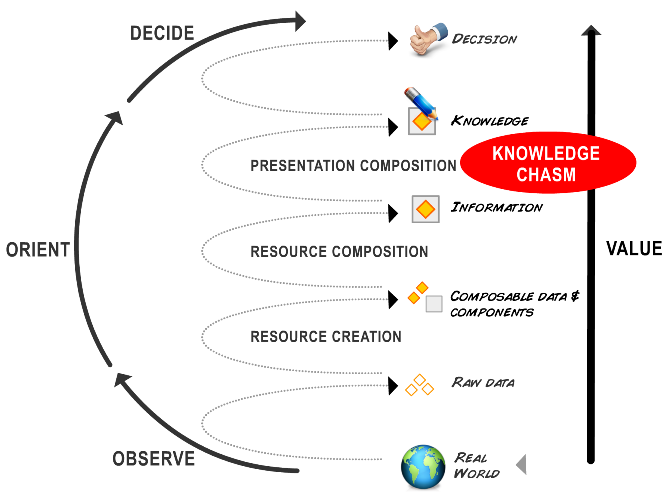

Figure 4, The Knowledge Chasm

In our model, users would have the capability to bridge this gap and stay in the same toolset while making the transition from information to knowledge. We envision this presentation composition as the capability to (1) compose text comments, graphics, audio and video into individual "pages", (2) string together pages to create full presentations, and (3) link from pages to the underlying information displays that were created in the *resource composition* step. The end result would look much like a slide presentation to the casual observer with graphics and bullet points, but to the creator(s), it would be a multi-layered tree of presentation pages and the supporting information. A similar approach was pioneered by the U.S. Navy's CTF-50 during Operation Enduring Freedom in Afghanistan. The fleet staff created an HTML-based "Knowledge Web" that allowed them to build information and present knowledge in the same tool, thus allowing them to dispense with slide presentations and the knowledge chasm.[8]

VII. Agility as the Driver of C2 Capability

All of this culminates in a view of C2 that directly parallels the OODA Loop of the command as the staff builds understanding (and knowledge) and the commander makes decisions as illustrated by Figure 5, Ecosystem with Roles. By making the creation of compositions parallel to the existing staff process, we believe that this system is both more natural and better able support staff operations. Moreover, we think that by having a one-to-one relationship between capability and staff activity, there will be a much-increased modularity in the system and an ability to swap out components (e.g., computing tools, widgets) without disrupting the entire value chain. The benefits of this modular strategy will be seen over time as capabilities are reused and evolved to meet new demands without the need of repeating the work of initial capability development.

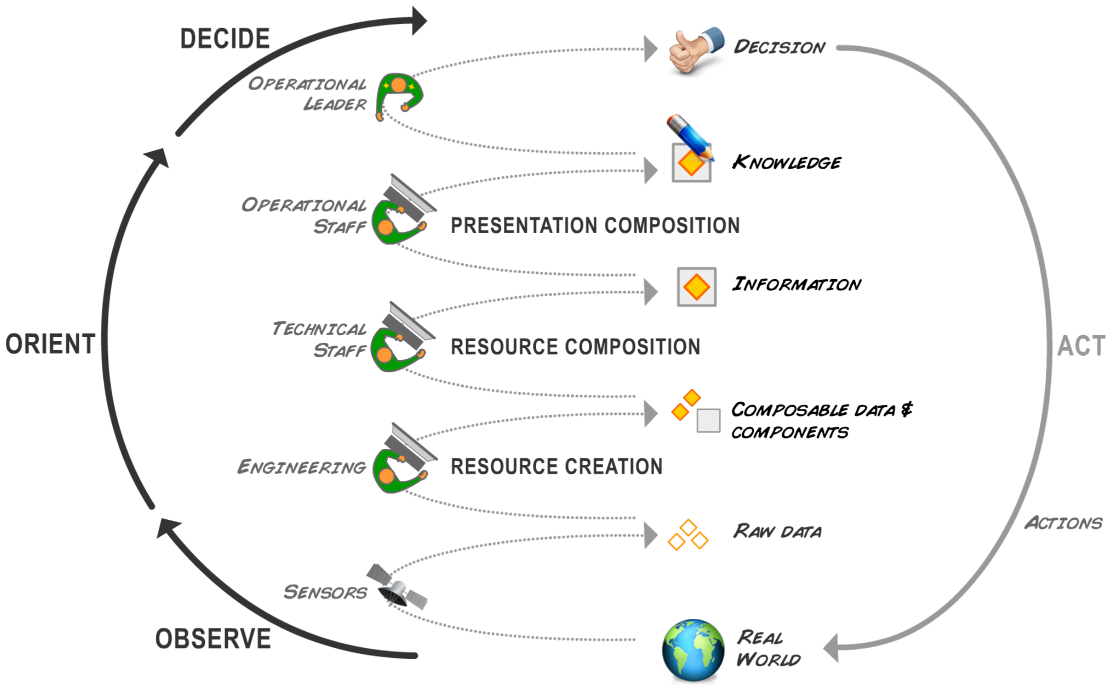

Figure 5, Ecosystem with Roles

This modularity also makes possible a much higher degree of C2 agility. As the staff is confronted with new knowledge requirements, they can easily compose, decompose and recompose information displays out of the component data and computing resources. One or more information displays can also be used and structured to support a higher-level knowledge page that is presented to the commander, upon which he/she can base decisions. All of this can be accomplished seamlessly in the same toolset and with much less effort and/or engineering support. Additionally, collaboration is better enabled as compositions, pages and presentations reside within a common system rather than in stovepiped information systems or on an individual's workstation.

VIII. Roles in the System

Overview

In the field of Human Factors Engineering, there is a common practice of creating notional "personas" to represent projected user types for a product or service. This method produces a point of reference that is vital in the planning, designing, and development processes by ensuring that a range of critical and diverging needs, as well as abilities, are addressed.

The remainder of this section defines the critical operational roles that are necessary to comprise a composition environment (See Figure 6, Composable Ecosystem Visualization). Note that it is not necessary for an individual user to be fixed in one of these personas as many individuals may vary from role-to-role depending upon context and task.

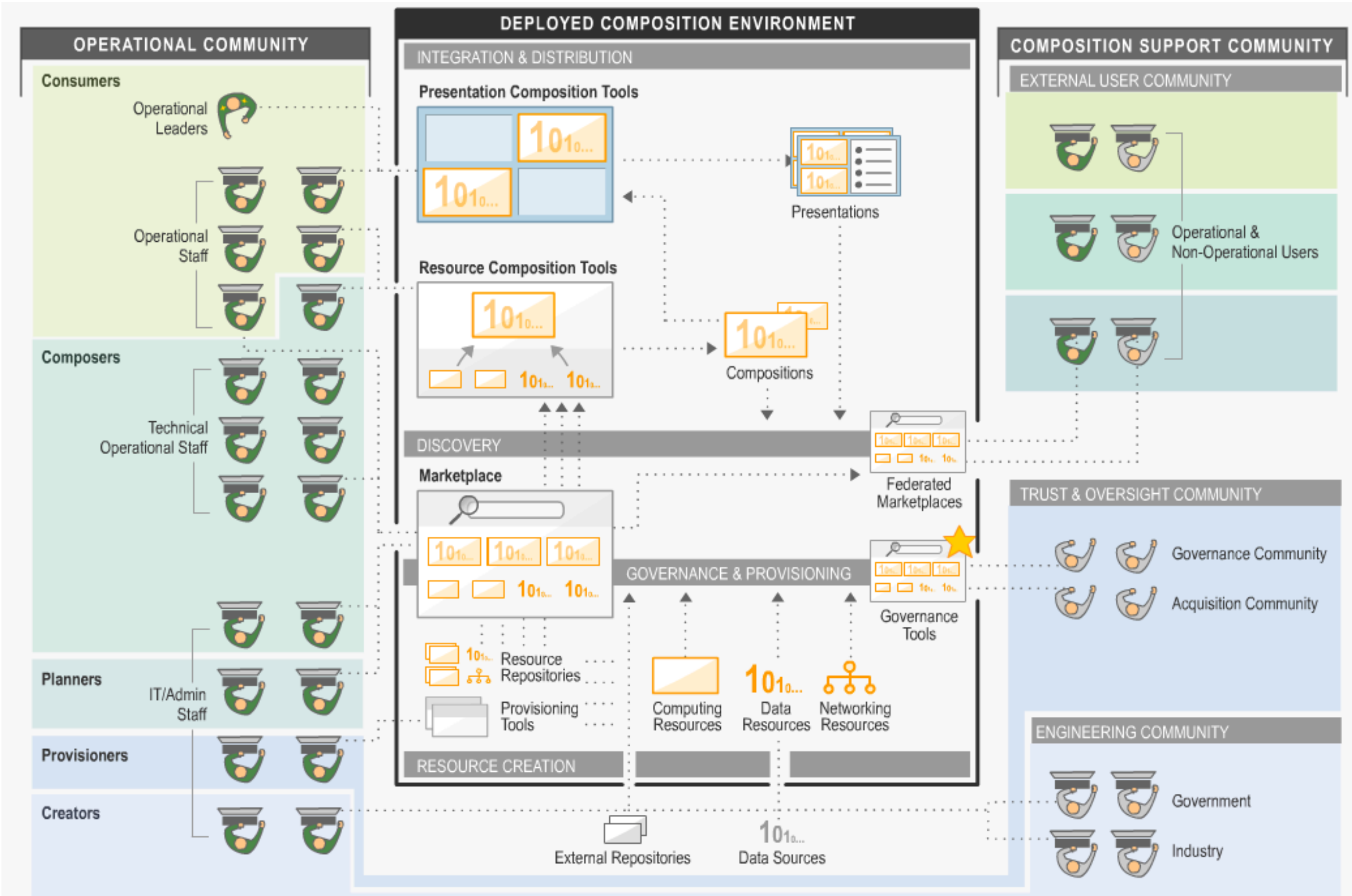

Figure 6, Composable Ecosystem Visualization

A. Persona 1: Operational Staff Member

The most common persona in the C2 environment is the Operational Staff Member. This notional persona includes common C2 roles such as Analysts, Operators, and similar positions. When assigned in a C2 environment, these are the primary users of the C2 systems-of-record. Their activities are highly data-centric and collaborative, likely splitting their time between C2 systems and more accessible tools that support collaboration, such as Microsoft Office and online chat.

Outside their C2 experience, the Operational Staff Member is likely comfortable with popular commercial software and web-based applications. For example, this notional person would most likely have experience with Google, Google Maps, web-based email, Microsoft Office, online chat, and popular websites, such as Amazon and YouTube. These experiences from the civilian world have a significant impact on the technology expectations for the operational staff member. These expectations will carry through from expected user interface behaviors to desired functionality and performance. The recognition of user expectations was a significant driving factor in the design of the CCOD marketplace that is discussed later in this paper. It is not coincidental that the marketplace functionality, interaction design, and information architecture aligns with patterns present in such popular industry marketplaces as Apple's iTunes, Netflix's movie repository, or Amazon.com. Aligning with user expectations will reduce training needs and minimize errors for users of the system.

To understand the implications of this persona, it is important to consider the behavioral patterns and common preferences that we observed of them. During extensive user studies, the Operational Staff Member consistently demonstrated a "get the job done" attitude that featured the willingness to step outside the bounds of the deployed systems-of-record to meet their responsibilities. Again and again, across a range of C2 environments, this tendency has led Operational Staff Members to adopt the tools they trust to help them do their jobs. These readily adopted tools often share the attributes of being easy-to-use, ubiquitous in the civilian world, and supportive of collaboration. Examples of tools readily adopted by the staff member include Microsoft Office, mIRC Chat, and Google Earth.

A notable factor in the behavioral patterns of C2 personnel is associated with the circumstances that surround their roles. They are often deployed for a period of time (e.g. 6-12 months) that is far too short to build true expertise with complicated software systems, even with a training program and dedicated support. When this factor is combined with the previously-mentioned "get the job done" attitude, it should not be surprising that staff members would not be motivated to learn a complex system if they can adequately satisfy their task by other means.

When designing tools and services for the Operational Staff Member, it would be most beneficial to consider the technology proficiency and expectations that they likely developed from their experiences in the civilian world. For example, if an application features data search capabilities, it would be wise to design a user experience and interface that is not unlike that of popular commercial online search engines, such as Google. The result is an interface that is easier to understand and navigate, which would likely lead to a reduction in user errors. By effectively and appropriately aligning with popular commercial user interface paradigms, systems may be more easily adopted, understood, and mastered. It may even lead to a reduced need for training, which is a considerable cost. The bottom line message here is simple: If you want your system to be adopted by the C2 community, lower the barrier of entry for operational staff.

The key theme in regards to designing for this role is simplicity. Note that this does not mean that functionality should be stripped from the system. Instead, the solution more often lies with organization and prioritization of the system's content and functionality. Well-designed interfaces give effective cues that help their users understand the structure of the system and how to navigate throughout it. In addition, these systems prioritize the most commonly needed information and features, and push less popular content down to lower (but accessible) levels. Measures such as this may allow a complex system to be perceived as easy-to-use, which is critical to adoption.

B. *Persona 2: Technical Operational Staff Member*

The second persona of note is the Technical Operational Staff Member. Sometimes this is an assigned and dedicated technical support role, but more often, this is a self-selected responsibility taken on by an operator with advanced technical abilities. Within the C2 operator community, they are the ones with the right technical skills to create technology-based workaround to software constraints or ad hoc solutions to emerging mission-related needs. While most C2 personnel are technically savvy with software and web application use, the technical staff member often has the knowledge of software programming languages, or is at least adept at creating advanced calculations and spreadsheet functions. Technical solutions provided by the Technical Operational Staff Member may range from complex Microsoft Excel spreadsheet creation to actual website or software development.

In today's reality, Technical Operational Staff are the unsung hero of the C2 environment, as they often provide "good enough" workarounds to capability gaps that arise too quickly for the traditional technology acquisition process to respond. While the Operational Staff Member requires an increase in simplicity, the existence of this technical persona suggests an entirely different need: adaptability. Through their actions, the technical staff members have shown the ability, motivation, and situational expertise to build their own solutions where they are needed. For that reason, it is recommended that the skills of this persona be embraced by building systems with the option for extending or adapting capabilities. Of course, it is critical that these advanced capabilities be appropriately positioned within the application in order to not interfere with its usability. If the functionality is intended for and valued by the Technical Operational Staff Member, they will be motivated to discover it.

Command-and-control processes are considerably complex. While ease-of-use and good design tenets are worthy goals, we must be realistic and realize that some tasks should and will be performed by the more technical personnel in a user base. In the case of C2 environments, we are fortunate that there are individuals up to this task. They are already taking on a composition role– we are simply recommending they be recognized, addressed, and enable in the development and design of C2 applications. By being willing to take on technical tasks, these technical staff members can actually create a simpler environment for their less technical counterparts.

C. *Persona 3: Operational Leader*

The next persona of note is the Operational Leader. This is an intentionally generic term as the persona may refer to anyone who is in a leadership or decision-making role within a C2 environment. The particular position may range from a cell chief, up through a joint force commander, and perhaps even the commander-in-chief himself. The intent with this persona is to define the individual for which the C2 activity is being conducted. More specifically, this is the person who is driving system requirements, consuming distilled information, and ultimately making critical decisions. In order to understand the C2 environment as a system, it is necessary to include the Leader as a significant factor.

An interesting aspect of the Leader persona is its influence. While they will be naturally small in number, Leaders are the driving force behind the core workflows of the C2 environment. Specifically, they sit on top (or within) a highly structured and well-organized chain of command. It is true that people in the commercial and academic domains do create reports and compose presentations for the consumption of others. However, their experiences most likely do not mirror the highly mechanized and fast paced chain of command demonstrated in daily C2 operations.

The consequence of this chain of command is that users are most often analyzing data or producing information for someone else's consumption at a higher level. For example, a given Operational Staff Member may utilize a software system to analyze incoming mission-related data. However, his or her ultimate role is to understand the implications of the data, distill it down to key data points, add some level of expertise-driven insight, compile that information into a Microsoft PowerPoint slide, and send that output "up the chain" to his or her leadership.

The implications of Leader-driven workflows on C2 system design are significant. All data-driven C2 systems should be tailored for users to easily and quickly produce digital documents and presentations, print physical reports, and enable user input. This will allow the chain of command activities to flow smoothly within the appropriate tools without the need of augmenting the workflow with manual user input into Microsoft PowerPoint and mIRC Chat.

D. Persona 4: IT System Administrator

Military operations are increasingly impacted by and tailored towards network-related activities. As a result, it is necessary to address the IT System Administrator as a primary role in the ecosystem. In regards to technical skills, this person may have similar capabilities as the individuals that would fall within the persona of Technical Operational Staff. The key distinction, however, is that the Systems Administrator plays only a supporting, non-operational role in the Command-and-Control workflow. Their primary responsibility is to ensure that the composition ecosystem is running as intended in support of operational activities. The IT Systems Administrator also plays a planning and provisioning function in regards to determining how resources will be managed and distributed within the composition environment.

E. Persona 5: Engineering Community

The fifth persona in this framework is that of the Engineering Community. This persona represents those that are actually developing the software resources to be composed. Engineers can reside in any number of places in the organizational network. With programs of record, they typically belong to a contractor hired to create the system or subsystem. They may, however, belong to the operator's organization in rare cases where development work is an organic function or where the organization has the requirement to innovate.

Engineers typically do not have direct intentional involvement in C2 workflows, but their work clearly influences the users within the operational environments. Moreover, they may inadvertently encode workflows into the systems they build. This can happen for a number of reasons – misunderstandings, unclear requirements, etc – but it usually boils down to the fact that the engineer rarely has the first-hand knowledge and experiences of an operator. Thus, when engineers build a system, they do it from particular vantage point and they make decisions based in part on their own needs and desires (e.g., ease of coding).

Engineers may also fill roles beyond strictly developing information technologies. They are typically tasked with updating software as users request changes or find problems and bugs. A growing role for engineers is that of systems integrator as it is very common for users to find value in the integration of data from different systems.

F. Persona 6: Governance & Acquisition Communities

The sixth and final role utilized in this persona framework is that of the Governance and Acquisition Community Member. While these are clearly two distinct communities serving unique purposes, they share a common goal of ensuring that the creation and management of resources are executed in an efficient and trustworthy manner. This role serves a critical function in policing and monitoring the quality of the composition system. This is, however, a delicate balance of control because the nature of a composition environment is such that the user community may ultimately control the course of the technical evolution.

IX. TECHNICAL CONSIDERATIONS

As mentioned above, one of the key attributes of this model is the way in which the building of the compositions mimics the traditional OODA Loop and how that affords the ability to have individual capabilities that map to natural functions. In this section, we discuss the attributes of specific capabilities within the ecosystem. Specifically, there are three tools that help to transform raw data to information in the value stack and a tool for sharing resources, compositions and presentations.

A. Resource Creation Tool

The lowest technical capability in the value chain is the *resource creation tool*. It will be used to access a nearly infinite range of data sources (e.g., SOAP services, spreadsheet files, text streams) and translate them to a common format that can be handled easily by the system's computing resources.

The most difficult challenge in building this tool is to make it simple and intuitive enough for use by non-technical people. As such, we believe that it should be focused on doing a very limited number of things well.

Resource Creation capabilities:

- Simple graphical user interface
- Select and access potential data sources
- Allow users to select columns, rows or ranges for output in the transformed data resource
- Combine two data sources to create a data resource based on a common key(s)
- Enact translation rules or services for fields (e.g., translate between GMT and local times)
- Format the output data

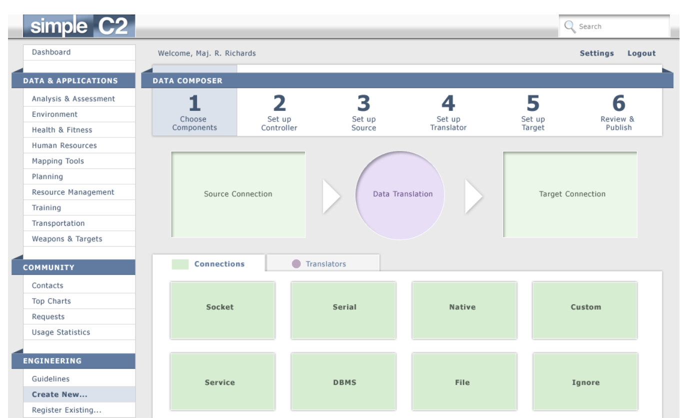

Figure 7, Resource Creation Concept Design

B. *Resource Composition Tool*

After the raw data sources have been transformed into data resources, the users then need a means for marrying them with display (i.e., widgets) in a GUI in order to create information compositions. This tool would also be limited by the need to maintain absolute simplicity for non-technical end users.

Resource Composition capabilities:

- Simple graphical user interface
- Select a page layout
- Select widgets to fit within the page layout
- Combine data sources for each widget
- Publish the completed composition
- Decompose existing compositions to change/update resources and/or settings

We believe that the library of widget types is finite, but includes the following: map, globe, timeline, data table, graph (i.e., node and edge), image viewer, and video/audio playback. Additionally, it is certain that specialized widgets (e.g., data input form) would emerge as the system evolves.

C. *Presentation Composition Tool*

The presentation composition capability is a simple one, but it has profound implications for workflows if users are able to bridge the "knowledge chasm" and continue on to create knowledge presentations in the same toolset.

Presentation Composition capabilities:

- Simple graphical user interface
- Build threads of individual pages that can be stepped through like a slide presentation
- Provide an editable text widget that can be included in pages so that users can add bullet comments, textual comments, etc.

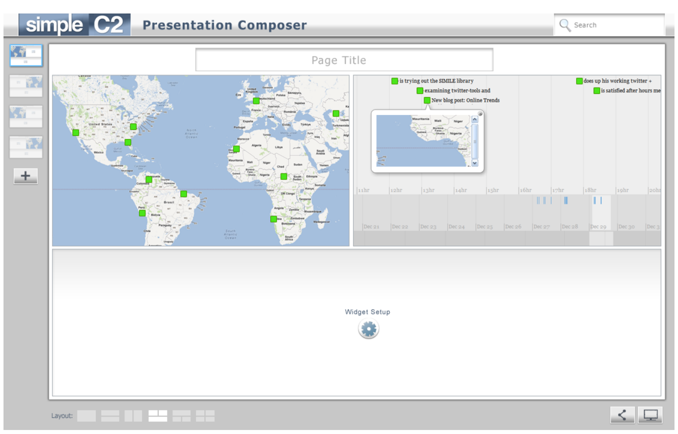

Figure 8, Presentation Composition Concept Design

D. *Marketplace*

The potential for the system also hinges on the ability of users to efficiently and effectively find resources, compositions and presentations that they need to perform their jobs. Additionally, the marketplace could provide a host of other services that would be useful to the user community.

Marketplace capabilities:

- Simple graphical user interface
- Exchange of resources, compositions and presentations
- Audit information services to provide insight into system use and trends
- Federation capabilities to allow marketplaces to cooperate and share information amongst themselves, thus allowing users to search

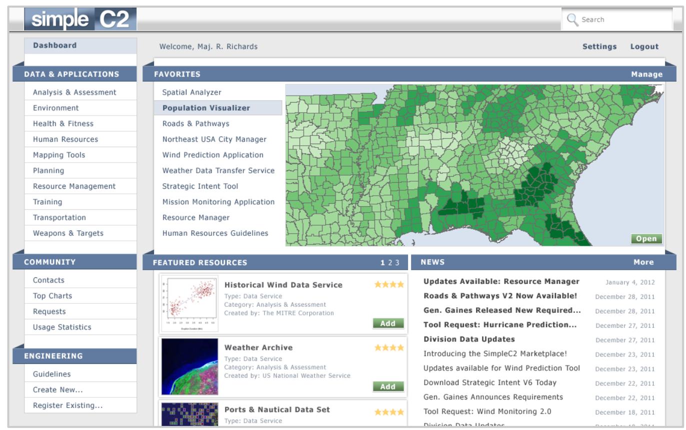

Figure 9, Marketplace Concept Design

Beyond these key capabilities, the system would also likely benefit from the following:

- Reputation/Rating Scheme – a capability for users to rate and leave comments on resources, compositions and presentations
- Resource Palettes – a capability for users to temporarily store resources on a "palette" that can be accessed in the presentation composition tool
- Requests Board – a capability for users to post requests for resources that are needed

E. *Other System Attributes*

There are also a number of other capabilities that have been identified that would provide

- URL Manager – a capability for assigning URLs to any resource, composition, presentation, user, or any other "entity" that may be referenced within the ecosystem. This would be useful in facilitating the handling of data and objects and in making references generic for end users.
- Workflow Manager – a capability for building simple workflows that can trigger compositions and actions by end users.
- Security – a capability to effectively and efficiently handle authorization and authentication with differing levels of access and permissions for different users.

- Pedigree – a capability to trace the origins and readily assess the quality of a resource.
- Vetting – a capability for users or user groups to place their own criteria for the approval of the creation and sharing of resources and compositions within the system.
- Means to express a new need or requirement to an acquirer or developer organization

X. Discussion

With the described set of tools and roles in place, we can begin to do some thought experiments as to how the composable ecosystem might work to promote C2 agility and improve the acquisition of information technologies.

The primary change that the acquisition and engineering communities would experience is that they would generally have to spend far less time and energy trying to derive and understand user requirements. Instead, their primary role would be to create, test, and accredit the raw materials (i.e., data resources, computing resources and widgets) that would fuel the development of new and/or improved compositions. Although initially threatening, such a scheme would likely be a better arrangement for all involved. With some engineering assistance, the users could build their own complex, but flexible solutions for their problems. These would evolve over time and would better keep pace with their environment and real needs. The acquisition and engineering communities could, in turn, focus on what they do well – building specific, hardened capabilities.

There would also need to be a fundamental shift in how needs are expressed and sensed by the ecosystem. Today, systems tend to operate until users have a problem sufficiently annoying to elevate through the chain of command. Often analysts are dispatched to gain a deeper understanding of the shortfall, which is then written up and submitted as an official requirement where it is ranked as a priority for funding. Little can happen without something being deemed an official, vetted requirement. As a result, response is slowed and only the most important issues tend to get addressed consistently. Perhaps more significant is the fact that the original intent of the need is often lost through the complex requirements development process.

In a composable ecosystem, users could begin to address their own needs. When they were immediately successful, the issue would end. If they continued to have problems, however, they might submit an entry to the *requests board*, that could be monitored by the acquisition and engineering communities. They might direct the user to an unknown existing resource or develop something to meet the need if it were sufficiently important. Beyond this kind of spot intervention, the acquisition community might also keep an eye on usage trends in order to act predictively or to find system improvements that might benefit the system that are not even being requested (e.g., the hardening of a widely used resource). In any case, the acquisition of IT could become more proactive.

Another place that would require significant change would be the certification and accreditation of resources and solutions. Currently, there is a fairly heavyweight and slow process for approving IT systems (or significant changes to systems) that must be accomplished before they are cleared for use in the field. To create a truly agile ecosystem, certification and accreditation would still need to take place, but it would tend to be for individual components rather than for whole groups of capability (as happens today). Much of the task of accepting or rejecting larger capabilities would fall to the commands and organizations actually using them. We imagine that a *vetting* capability could be used to control the use and sharing of resources and compositions within an organization. For instance, a given command might set up business rules that would allow individual users to build the compositions they desire, but once they wanted to share a solution with another user in the command it would need to be approved by an authority. Furthermore, there might be ever more stringent approval requirements as a composition were shared more widely (e.g., between commands) or was included in a critical workflow. Although such a scheme is a large departure from the status quo, we believe that it is entirely fitting that the experts that use and are ultimately responsible for an operational thread should also be the ones to say whether a solution works reliably and efficiently enough. After all, this behind-the-scenes approval scheme is already happening with the many brittle workarounds that commands build everyday in Microsoft Excel and other tools to get the job done.

Additionally, we might experience more in the way of standardization among organizational use. This may seem counterintuitive in the face of a capability that could entertain nearly infinite variation. In the real world, however, what we tend to see with respect to evolving ecosystems is that successful variations are often widely adopted and become de facto standards. Again and again on the Internet, we see the emergence of use patterns best described by a power law distribution – whereby in any given population, a very small number of individuals (e.g., websites, search engines, tools) receive a really disproportionate level of use.[9] So today, an innovative workaround built in one command can't easily compete with others and can't be easily shared. With less friction for creation and sharing, we might actually see more agreement and adoption of the best-of-breed solutions rather than a settling on whatever is locally available.

XI. Conclusion

Eventually, with composable capabilities in place, the end user community would experience the benefits of a C2 environment that is more responsive to change and consistently aligned with its needs. Solutions to emerging problems would be formulated much more quickly. Innovations could be shared much more easily and we could expect that there would be marked improvements in effectiveness, efficiency and speed. We think, nevertheless, that the greatest improvements would come from three attributes of such an ecosystem. First, the end users would "have the stick"; they would be the ones directly shaping C2 information solutions. Where today warfighters have many proxies (e.g., acquisition community, requirements generators, test community) representing their interests in the development of new capabilities, under this scheme the end users would represent themselves. They would be the ones directly engaging in the trade space to find the best solutions and the various support groups could recede into the background – where they belong. Second, a composable ecosystem could support a nearly infinite range of user needs. This ability to come up with a best-available information solution for any given task will have broad implications for real-world operations. Finally, the C2 agility of such an ecosystem would be head and shoulders above our current systems approach. A force with such agility would be able to rapidly build tailored solutions that would aid in moving up the value chain from data to knowledge to decisions much more quickly than an adversary that is not similarly equipped.

XII. References

[1] "Major Weapon Systems Continue to Experience Cost and Schedule Problems under DOD's Revised Policy," U.S. Government Accountability Office, GAO-06-368, Apr. 2008.

[2] "1961-5-25 Address Before a Joint Session of Congress." [Online]. Available: http://www.jfklibrary.org/Asset-Viewer/xzw1gaeeTES6khED14P1Iw.aspx. [Accessed: 06-Feb-2012].

[3] Committee on Improving Processes and Policies for the Acquisition and Test of Information Technologies in the Department of Defense; National Research Council, *Achieving Effective Acquisition of Information Technology in the Department of Defense*. Washington, D.C.: The National Academies Press, 2010.

[4] "Department of Defense Policies and Procedures for the Acquisition of Information Technology," DEFENSE SCIENCE BOARD WASHINGTON DC, DEFENSE SCIENCE BOARD WASHINGTON DC.

[5] J. Gall, *Systemantics: How Systems Work and Especially How They Fail*. Quadrangle, 1977.

[6] F. Osinga, *Science, Strategy and War: The Strategic Theory of John Boyd*, 1st ed. Routledge, 2006.

[7] J. Boyd, "The Essence of Winning and Losing," 28-Jun-1995.

[8] M. Adkins and J. Kruse, "Case Study: Network Centric Warfare in the US Navy's Fifth Fleet," *Research Report for the Office of Force Transformation, University of Arizona Center for the Management of Information*, 2003.

[9] R. Albert and A.-L. Barabasi, "Topology of evolving networks: local events and universality," *Physical Review Letters*, vol. 85, no. 24, p. 13, 2000.