2010 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
VEHICLE ELECTRONICS AND ARCHITECTURE (VEA) MINI-SYMPOSIUM
AUGUST 17-19 DEARBORN, MICHIGAN

TIMING IS EVERYTHING AND A PICTURE IS WORTH A THOUSAND WORDS

Alan Woody
Raytheon
Plano, TX

Curtis Hoff
Raytheon
Plano, TX

ABSTRACT

The U.S. military has made substantial progress in developing and fielding C4ISR systems that can collect and gather overwhelming amounts of valuable raw sensor data. A new challenge that has emerged with the deployment of numerous state-of-the-art ISR collection systems is the effective and timely use of the collected surveillance and reconnaissance information, or simply stated an architecture that pushes the timeliness and accessibility of this situational awareness data to the tactical edge – "the right data at the right time to the warfighter." Along with this information distribution challenge is the increased size, weight, and power implications of the numerous stove piped systems that are bolted on to the mechanized platforms. The tactical plug-and-play framework integrates cohesive, yet loosely coupled infrastructures for communications, Command and Control (C2) applications, sensor suites, and provides the "digital backbone" architecture. This architecture results in a configurable, single-screen interface for the operator to monitor and control all systems integrated onto a vehicle, and a reduction in the space claim required for operation of these systems.

INTRODUCTION

Delivery of real time key information to include situational awareness to a decision maker is what makes the difference between loss and victory on the battlefront. The challenge of getting "the right data at the right time to the warfighter" is heightened by the lack of high bandwidth communication connectivity between platforms and the numerous stove-piped systems that are deficient in interoperability characteristics. The resultant vehicle architecture is a collection of stand-alone radios and systems, each requiring their own displays and input devices. This ad-hoc architecture results in two problems: the need for additional - more capable systems to produce and collect data that result in even more interoperability and weight/space problems, and additional workload associated with the increased information overload for the operators of the system.

The objective of this paper is to present an architecture framework with the following characteristics:

- Scalable - so that the architecture supports a wide variety of equipment packages and mission variations
- Interoperable – provides an architecture that can accommodate *multiple* existing system packages and consolidates command and control for multiple mission profiles (i.e. surveillance, situational awareness, etc.)
- Usable – provides a logical, intuitive easy to use interface at all user levels (i.e. war-fighter, battalion commander, etc)
- Effective – increases utility and mission effectiveness of the systems on the platform by providing the necessary data, to the right place, at the right time, for a wide variety of missions, resulting in increased survivability.
- Provide the above features with reduced size, weight and power footprint on the platform

The Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) systems onboard most combat vehicles are currently co-located but operate independently. The independent nature of the C4ISR package (i.e., its physical footprint) has a significant impact on overall size, weight, power and ergonomics. The Tactical Plug and Play framework provides a combination of hardware and software systems that are interconnected in order to allow centralized access to sensors, command/control, and situational awareness information, as well as system physical assets such as radios, and the distribution of real or near real-time video data to users. This integrated approach maintains or enhances operational capability and interoperability, and improves ergonomics while reducing size weight and power.

The purpose of the Tactical Plug and Play framework is to consolidate command and control of the multiple systems utilized in support of a wide range of military operations, such as surveillance, communications, collection of enemy order of battle information, Battle Damage Assessment (BDA), battlespace management, situational awareness (SA), and targeting support.

This paper will present the Tactical Plug and Play framework architecture that has been developed to solve the scalability and usability problems. Another focus of this paper is the Size, Weight, and Power (SWaP) concerns introduced by adding more disparate systems to platforms and how the Tactical Plug and Play architecture alleviates this problem. Finally, this paper will show that the Tactical Plug and Play framework reduces mission execution timeline to increase mission effectiveness and survivability.

THE NEED

The transformation of the Army from heavy forces to lighter, faster forces increases the burden on today's video intensive C4ISR systems to maintain the lethality and survivability of the current forces. The objective of this effort is to develop a scalable architecture that integrates surveillance and reconnaissance, active protection, combat identification, and communication capabilities within the reduced space of platforms and dismounted warriors. It is also to provide the same look and feel to the warrior regardless of whether the warrior is operating in a simple mechanized platform, a Reconnaissance, Surveillance, and Target Acquisition (RTSA) platform, or in a mobile command post. The framework expands the warfighter's capabilities and increases the effectiveness to perform disparate missions, while reducing the mission execution timeline. Figure 1 illustrates the comparison of the capabilities of today compared to the efficiencies provided by the framework. The efficiencies are evidenced by the shorter mission times indicated by the shorter timelines to reach the End of Mission (EOM) criteria. The digitization of command and control functions and distribution of video data allows the warrior to reduce the amount of voice required during missions, which increases the survivability through decrease threat exposure. However, the network continues to support voice traffic along with the ability to digitize audio to be associated with the other collected situational awareness data and information.

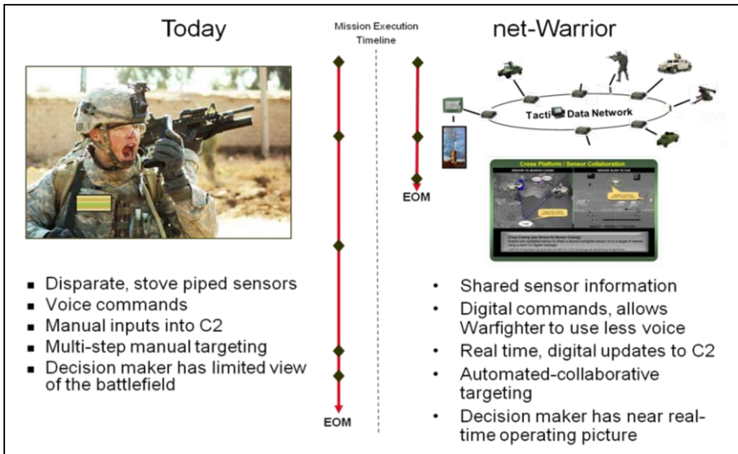

Figure 1: Capabilities to the Tactical Edge

Finally, this architecture is designed to enhance the tactical network and provide timely situational awareness information to the warfighter. Currently, there is a gap of capabilities to provide timely situational awareness information all of the way to the tactical edge. There is a fairly well defined flow of information from the Company level to the Battalion, Brigade, and higher, but there is limited flow of information and knowledge, in a timely manner, to the mechanized platforms and below. Figure 2 delineates this gap of capabilities and information flow.

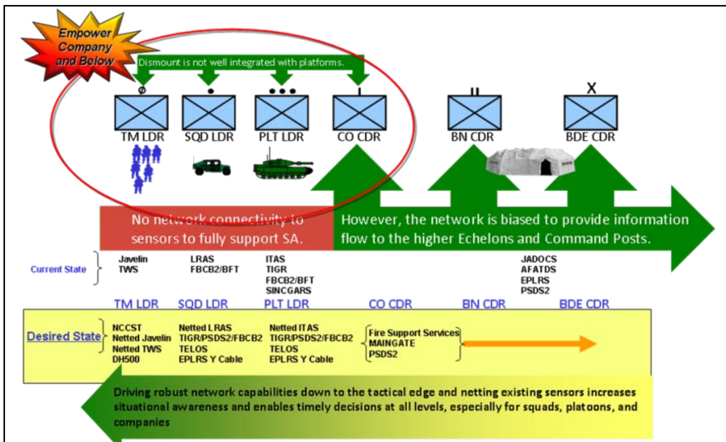

Figure 2: SA Down to the Tactical Edge

The Tactical Plug and Play framework integrates cohesive, yet loosely coupled infrastructures for communications, C2 applications, and sensor suites along with providing a digital backbone architecture. The Tactical Plug and Play framework is a tactical open standards-based C4ISR system integration architecture designed to be robust, flexible, and scalable, while reducing overall system weight and cost versus conventional implementations. The architecture integrates heretofore stove-piped, discrete systems into a common framework that enables sharing of information between users on the same platform and between platforms over a tactical network. This architecture results in a configurable, single-screen interface for the operator to monitor and control all systems integrated onto a vehicle, and a reduction in the space claim required for operation of these systems.

THE SOLUTION

The electronics architecture is comprised of sensing capabilities, which are integrated into a platform-centric command and control infrastructure, along with a communications backbone providing the connectivity between warriors, platforms, and operation centers/command posts. This integrated architecture has to be flexible and configurable to support a variety of mission or operational tasks. The architecture has to support the appropriate capabilities required to enable the warrior to execute the various missions.

Operational Tasks are partitioned into a hierarchical set of tasks, where each set is associated with an Operational Domain. An Operational Domain is defined as one of the following: Net-Centric Communication, Mission Planning, Netted Lethality, Assured Mobility, Situational Awareness, Survivability, Sustainment, and Training, where the first six are collectively considered Combat Operations. This task structure:

- provides a logical division of sensor tasks that are used to support military missions;
- represents a top-down organization that transitions from general behavior to specific tasks;
- provides a basis for behavioral, functional, and performance analysis;
- provides a basis for construction of decision trees that guide employment of sensor assets in the field, and defines sensor fusion approaches;
- is platform-independent;
- is technology-neutral to the degree possible;

Association of some of the tasks to the Sensing Domains is not always unique or exclusive. It should also be noted that the Sensing Domains are not totally independent of each other. In the battlefield, they support and interact with each other. An example is the close relationship between Situational Awareness and Lethality (i.e., you have to find the threat to neutralize the threat), and between Assured Mobility and Survivability (i.e., you have to be able to outrun/out maneuver the threat in order to survive).

The architecture described in this document has to address the various mission scenarios and associated capabilities within each of the operational domains, while considering the capabilities needed from a command post view, a mechanized platform view, and a dismounted warrior view. The resultant architecture has to be scalable and provide the essential capabilities for each of the instantiated viewpoints (mobile command post and various mechanized platforms).

Figure 3 shows the conceptual architecture for the Tactical Plug and Play Framework. The framework is a layered architecture with several integrated plug and play frameworks embedded in each of the layers. The embedded frameworks, facilitated through middleware, provide the flexibility and scalability attributes for the overall concept.

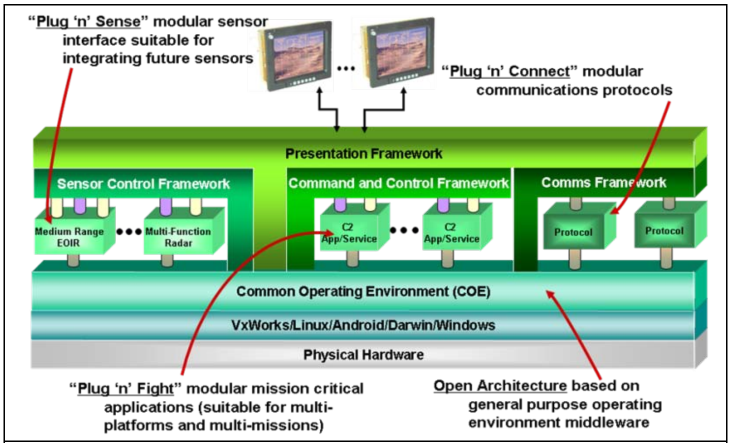

Figure 3: Architecture Concept

Commercial off-the-shelf (COTS) products and standards will be a heavy player moving forward. The architecture developed for the Tactical Plug and Play effort leverages open architecture concepts developed in the commercial environment. For example, the Google Android is a front runner for the small handheld, user-friendly device solutions space. Each one of the frameworks within the Tactical Plug and Play is architected with the same concept of the Android Application Framework in that the framework needs to provide capabilities that any application can leverage to implement its own functionality. The frameworks within the Tactical Plug and Play architecture are:

- Presentation Framework
- C2/Application Framework
- Sensor Framework
- Communications Framework

One of the main architectural patterns that comprise the Tactical Plug and Play Presentation Framework is the Model-View-Controller (MVC) pattern. The MVC pattern isolates "domain logic" (the application logic for the user) from input and presentation (GUI), permitting independent development, testing and maintenance of each. The model is used to manage information and notify observers when that information changes. The model is the domain-specific representation of the data upon which the application operates. The implementation of the model will reside in the Application Framework. The view renders the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes. A viewport typically has a one to one correspondence with a display surface and knows how to render to it. The view portion of the MVC is the piece that resides within the Presentation Framework. The controller receives input and initiates a response by making calls on model objects. A controller accepts input from the user and instructs the model and viewport to perform actions based on that input. The actual interaction with the user will be part of the presentation framework, but the interpretation of the user's input, which is the primary function of the controller, will reside in the Application Framework and the Sensor Framework. The benefit of the MVC pattern is that the view and control components are independent from the model and each can be independently modified without impacting the other. For example, the control piece can change the type of input supported without impacting the view or model aspects. This approach also allows for multiple views to exist for a single model, which supports different user interface devices.

The main concept embodied in the Sensor Framework is that there are several types of interconnects that have to be supported due to varying interface requirements; such as high bandwidth video, low-latency command and control, and high-level command and control interactions. The Sensor Framework will interface directly with the Presentation Manager for the low latency and display video interconnects and it will interface with the Sensor Manager within the Application Framework for the high-level command and control and video storage interconnects.

One of the main concepts exploited within the Tactical Plug and Play architecture is that of an Integration Architecture. An Integration Architecture covers the communication technologies and the interaction between different systems and the applications, processes, or threads, within a system. There are two main types of communication architectures employed in the Tactical Plug and Play architecture; namely, the Publish-Subscribe Messaging paradigm, and the Service-Oriented Architecture (SOA) paradigm. The real-time control portions of the architecture, such as the Sensor Framework, will exploit current and future communication technologies to make communication between components/capabilities flexible and scalable, while simultaneously making the system more usable to the war-fighter. The Application Framework and Presentation Framework will exploit the SOA capabilities, such as Web Services and an application server approach, such as JBoss. The layered architecture in each framework demonstrates the importance of the middleware and infrastructure required to support the plug-in architecture, and provides the ability to easily extend and modify the functionality of each framework independent of the other frameworks.

Figure 4 illustrates a wire diagram that represents the components comprising the high level architecture framework within a platform. As illustrated in the diagram, the architecture allows for the number of sensors and communication connections to be configurable and expandable. Note that the C2 applications will scale appropriately to whatever missions are assigned to the platform and sensor complement installed on the platform. In addition, the communications architecture supports multiple radio waveforms to be supported independent and isolated from the applications that need to communicate over the IP enabled network. The architecture enables a scalable solution capable of supporting a range of mission configurations from minimal (i.e. single functionality) to intermediate (i.e. multi-role functionality C2 with EO sensor functionality) to complex (i.e. multi-role functionality such as reconnaissance, surveillance and target acquisition using multiple sensors).

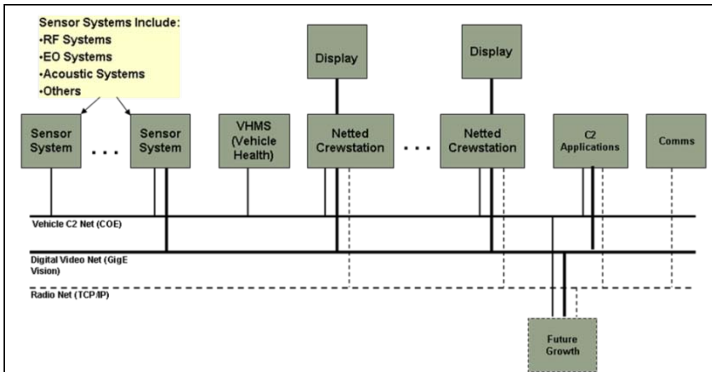

Figure 4: Platform Scalable Architecture Concept

Figure 5 depicts a reasonably complex instantiation of the platform framework for a reconnaissance and surveillance class platform with multiple sensors integrated into a suite that provides coordination and cross-cueing on the same platform. The complexity associated with this platform reflect the multiple mission capabilities required, such as surveillance – target detection and tracking, indirect fire control, BDA, as well as direct fire engagements. This instantiation is also representative as it comprehends a typical mission commander's role – coordinating the activities of others in the engagement.

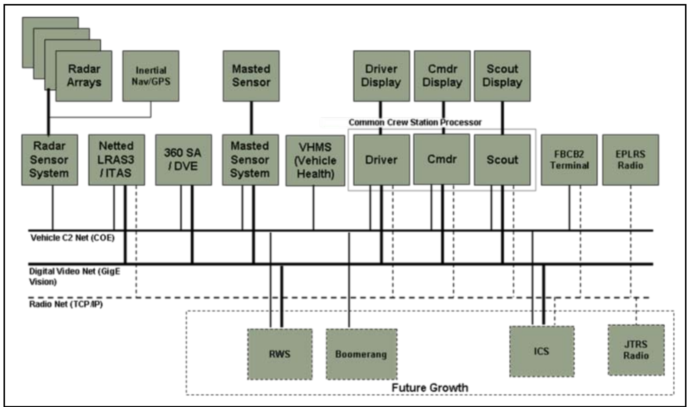

Figure 5: Complex Instantiation of Platform Architecture

The Frameworks

The Application Framework is comprised of several plug-in architectures; namely, the Communication Manager, the C2 Server, the Sensor Manager, and the Map Server. The Communication Manager, C2 Server, and Sensor Manager plug-in architectures are similar in that they all provide an application server infrastructure while the applications are the plug-ins that provides the functionality and APIs for external consumption. This is somewhat similar to the Android environment referenced above where the applications are fairly independent of each other but they all reside, and coexist within the application infrastructure. The benefit of this type of architecture is that the applications can be developed and deployed independent of each other and the applications can be deployed based on mission parameters and objectives. However, the Map Server plug-in architecture differs from these in that it provides the infrastructure for the plug-ins that deal with the formats and protocols particular to various map servers, such as Google and World Wind. The benefit of this approach is that the map implementation is hidden from the presentation framework and the different map implementations can be chosen based on performance and esthetics.

This approach enables the three frameworks to evolve independently in order to focus the Presentation Framework on the ease of use aspects while providing an infrastructure within the C2 environment that enables ease of modification and the addition of new capabilities.

Figure 6 depicts the importance of the infrastructure and middleware layers for the Application Framework. The middleware isolates the applications from the hardware and operating system specifics and details while providing an environment that enables easy deployment to one or more processors. The communication between the applications relies on middleware capabilities, such as publish/subscribe messages, which facilitate the delivery of information independently of the hardware and network topology. The diagram illustrates the ability to separate the different plug-in implementations from each other and deploy them on separate processors, or as separate processes on the same processor.

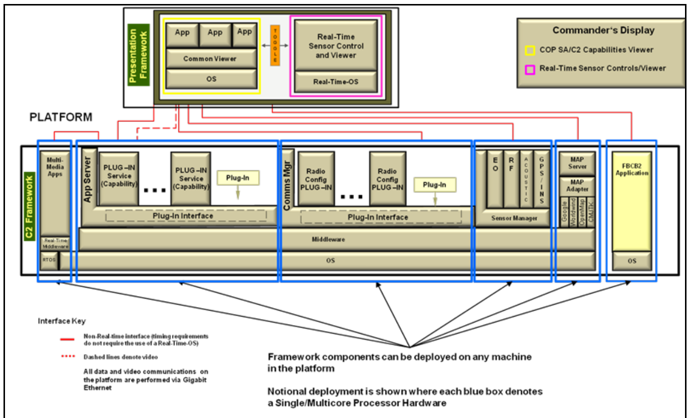

Figure 6: Application Framework and Middleware Role

This approach illustrates how legacy systems can interoperate with the Application framework through the use of the middleware capabilities. The messaging capabilities of the middleware facilitate the use of an adaptor to translate from one message set, and associated protocol, to another – (making two disparate systems common from a C2 point of view). In addition, the Presentation Manager provides the capability to access the legacy system through either a remote desktop type of window, or as an application that provides a stream of data to a view portal.

The Sensor Control Framework depicted contains the Sensor Framework Manager, which provides the coordination and low-level sensor resource management capabilities. As illustrated in Figure 7 the Sensor Framework requires the use on an RTOS to meet the performance requirements, and the goals of the performance quality attributes. Each Sensor class has a proxy that interfaces with the actual sensor hardware and provides a common and consistent interface to both external clients and the Sensor Framework Manager.

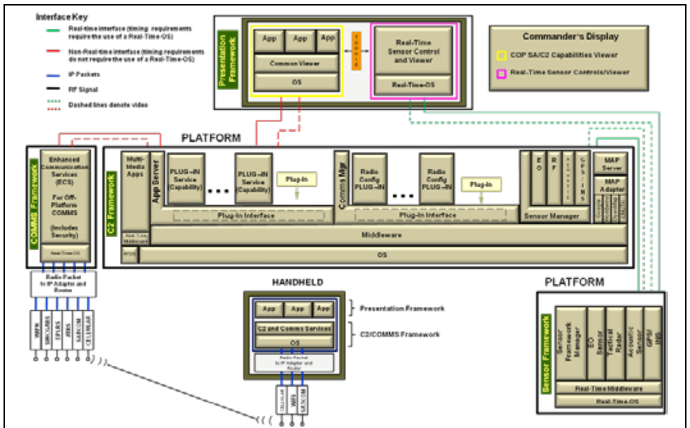

Figure 7: Frameworks

There is a low-latency connection between the Sensor Control Framework and the Presentation Framework. This is needed to meet latency requirements for closed-loop operations between operator input and sensor response that include video and other types of data. The diagram also depicts that the sensor control and video display are from the Sensor Control Framework to the Real Time Viewer capability within the Presentation Framework. This type of capability within the Presentation Framework is needed to address both performance requirements and usability concerns (like expected response times).

The interface between the Application Framework and the Sensor Control framework facilitates capabilities such as configuring the sensors, initiating search operations, and initiating cross-cueing between sensor types (such as RF to EOIR). The video connection between the Sensor Control Framework and the Application Framework supports both the fusion processing and the ability to store/archive video for playback and radio transmission.

When a new sensor is added to the sensor control framework, a plug-in has to be developed for the Sensor Manager component within the Application Framework in order to facilitate configuration, coordination and platform level sensor resource management. The plug-in architecture within the Application Framework is independent from the Sensor Control Framework, but there is a dependency between the individual plug-ins and the sensor components within the Sensor Control Framework.

The Sensor Framework Manager is architected with a type of plug-in architecture, but instead of sensor control, the plug-ins provides coordination control between sensor types.

Figure 7 also depicts the interfaces and capabilities within the Communications Framework. As illustrated in the diagram, the enhanced protocols, such as Disruptive Tolerant Networking (DTN), and security capabilities are provided within this framework. The configuration of the radio networks are provided within the Communications Manager element in the Application Framework.

A big part of the functionality provided by the Application Framework is the management of the information from various sources, such as friendly force location, enemy force locations, environmental information (such as weather and terrain), orders and plan information, and resource management. The totality of this information is managed and presented to the user, through the Presentation Framework, to provide something like a Common Operating Picture. This information is generally provided through a map interface to provide context.

As delineated in the Figure 8, the Sensor Framework Manager is a Plug-in architecture that allows for additional sensor components to be added or modified without impacting the existing sensor control plug-ins. The Middleware layers provide the isolation from the transport media, the processing hardware, and the required real-time operating system.

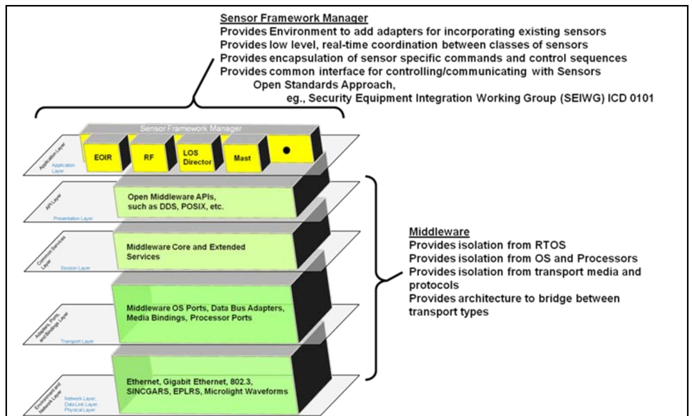

Figure 8: Sensor Framework Manager

The Communications Framework has to support various radio and network topologies, including legacy radio equipment, such as SINCGARS. The Communications Framework has to provide bridge and gateway capabilities to provide interoperability between the various radio waveforms and protocols. In addition, the framework has to incorporate and provide the latest communication networking capabilities, such as Wireless Fidelity (Wi Fi) and cellular communications. However, one of the areas that will have to be addressed with this new technology is the Information Assurance and Security areas that come along with the new technologies.

CONCLUSION

A plug and fight architecture for existing and future Army platforms has been presented. It achieves the objectives of scalability, interoperability, effectiveness, and usability while providing a reduced SWaP footprint relative to today's video intensive C4ISR systems. The architecture is comprised of four frameworks (C2, Sensor Control, Presentation, and Communication) that provide application server infrastructure while the applications are the plug-ins that provide functionality and APIs for external consumers of data. This approach, facilitated by middleware, allows independent evolution of the frameworks as well as the applications that give the system its capabilities. Another important benefit of this architecture is it is a digital framework that helps eliminate manual steps, which reduce mission timelines. Finally, this digital architecture provides the ability to accommodate legacy systems as well as the current and future generation capabilities which increases effectiveness by providing the necessary data to the right place and the right time for a wide variety of missions.