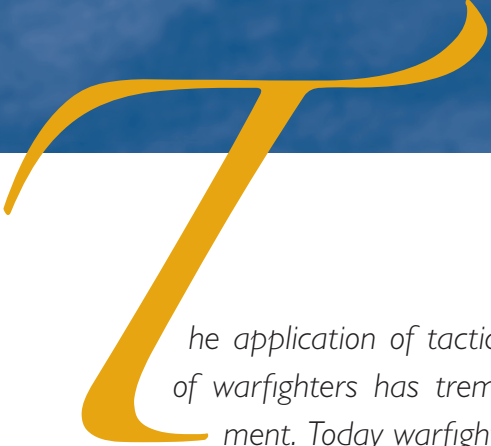


Tactical Spacecraft Commanding Service Architecture

*Puck-Fai Yan, Matthew J. Potter,
and James A. McGovern*



The application of tactical spacecraft or satellite constellations in support of warfighters has tremendous benefit in a dynamic battlefield environment. Today warfighters must queue behind a long list of higher-priority requests and wait for scarce opportunity to access space-based assets. By the time the information collected becomes available, it is often obsolete and has little tactical value. The effectiveness of ground spacecraft operations to respond quickly in a high-tempo, high-intensity situation is limited. A missing element to the success of these dynamic operations is the timely access, tasking, planning, and commanding of space-based sensors. The Tactical Spacecraft Commanding Service Architecture (TSCSA) emerged from an independent research and development effort to develop a prototype capability that will explore a fundamentally new way to access, task, and receive information from tactical spacecraft assets; it is based on the notion of a distributed, semiautomated planning and scheduling framework built using a service-oriented architecture and current Web-service standards. The TSCSA will present tactical spacecraft sensing assets as an abstract service to potential warfighting clients.

INTRODUCTION

Actionable intelligence requires timely information gathering and dissemination to warfighters. This need can be accomplished through the use of rapidly deployable assets from space, air, and ground. The Office of Force Transformation highlighted its need for rapid deployment and employment of new capabilities to improve the operational response to potential threats. As part of its five goals, the Office of Force Transformation has stated

net-centric warfare as a transformational element in the success of DoD defense strategy. The ability to efficiently and effectively use space systems and provide access to them at the lowest tactical level possible is of particular interest and concern. The primary space-system aspects that enable such capabilities include tactical spacecraft, the implementation of distributed planning and tasking (e.g., tasking requests to the satellite constellation as

opposed to a spacecraft; effecting distributed command and control), intrasatellite constellation communications, and distributed reporting. In parallel, DoD efforts in development of net-centric technologies within the Global Information Grid (GIG) are moving to radically new approaches for command and control of both tactical and strategic assets, based on the notion of service-oriented architectures that allow users to discover, access, and use new functionality during system operation.

Past research efforts have focused on the development of autonomous control and coordination for spacecraft constellations, with an emphasis on diagnostics, fault protection, and common modular architectures for both flight and ground software. These efforts have not specifically addressed the integration of tactical space sensor assets into the emerging service-based architecture being developed by the Office of the Secretary of Defense for Networks and Information Integration for the next-generation warfighting using the GIG. These efforts also have not addressed the problem of distributed spacecraft tasking (e.g., multiple clients independently requesting remote sensing services from a limited set of assets).

Innovative processes and ground systems have been developed at APL in civilian space applications that help us avoid operational fatigue and mistakes on the most challenging missions. In civilian space applications, the primary goal of our processes is to capture and execute the objectives defined by science team users. Scientists often command the spacecraft or instruments at a high level via these new processes and ground systems so that scientific goals are met without translation. In military space applications, the goal is to capture and execute the objectives defined by military commanders in much the same way. In FY2006, a team of APL engineers began an independent research and development (IR&D) effort to develop a radically new military space tactical spacecraft-commanding capability that focused on warfighter needs through the use of a service-oriented architecture (SOA). Figure 1 depicts a notional system architecture of such a system. As such, the assets will be discoverable, accessible, and understandable through standard Web-service mechanisms (e.g., Universal Description, Discovery, and Integration Registry and Web Services Description Language).

CONTEXT-BASED TACTICAL SPACECRAFT COMMANDING

The lifetime of tactical spacecraft may be much shorter than that of current satellite systems, and if responsive space concepts become a reality, then tactical, nearly off-the-shelf systems may be launched on short notice. The new paradigm of short notice and the desire to move execution capabilities farther out into the field will have several important operational consequences that have been addressed by this internally funded research effort.

Shorter useful spacecraft life means fewer opportunities to get the most benefit out of a tactical spacecraft, making each day more critical. As a simplistic example, consider that a \$30 million tactical spacecraft that lasts 6 months has roughly the same cost per day as a \$500 million system that lasts 8 years. If you take away 3 weeks from each system for launch and checkout, the tactical spacecraft's cost per useful day exceeds that of a traditional satellite; in addition, the tactical system simply offers fewer useful days to accomplish established goals. So the economics of short-lived systems demand streamlined operations. Attaining streamlined, highly automated, and easy-to-use ground systems lines up well with the desire to provide commanders in the field greater access to space resources.

One of the practical consequences of pushing access to spacecraft resources farther out into the field is that more people with less training will interact with tactical satellites; more people means a greater degree of coordination is required between the mission operations center (MOC) and warfighters. Without good processes and ground systems in place, coordination and insufficient training can quickly lead to fatigue and costly mistakes.

Context-Sensitive Commanding

When using our process, the commander or warfighter in the field does not need to know the internal workings of the spacecraft or sensors to discover what services it can provide and when those services are available. This approach reduces the need for training spacecraft field personnel on each payload of each spacecraft system. Our concept calls for a brief self-study training course including scenarios that execute at a realistic pace, followed by mentoring during actual routine operations.

As each new tactical resource comes online, a dictionary of capabilities is distributed to warfighters and the MOC alike. With the capability dictionary and common ground system in place, both MOC personnel and warfighters can submit speculative target requests to search for tangible opportunities to image their targets by using the menu of available capabilities. In some cases multiple remote warfighters express their priorities and provide opportunities in rank order to the MOC, after which the various plans are integrated and approved for execution. This approach enables the MOC to deconflict plans from multiple warfighters, combining opportunities when possible and in some cases moving opportunities to alternate orbits. Our architecture does not specify the method of integration or how the needs of the different warfighters are prioritized. Once the observation plan is approved, the spacecraft-specific commands for the observation are automatically generated, scheduled for upload, and delivered to the appropriate MOC. The basic use cases of ingesting a capability dictionary, ingesting a current schedule, searching for observing opportunities, and generating commands would be the same for each new

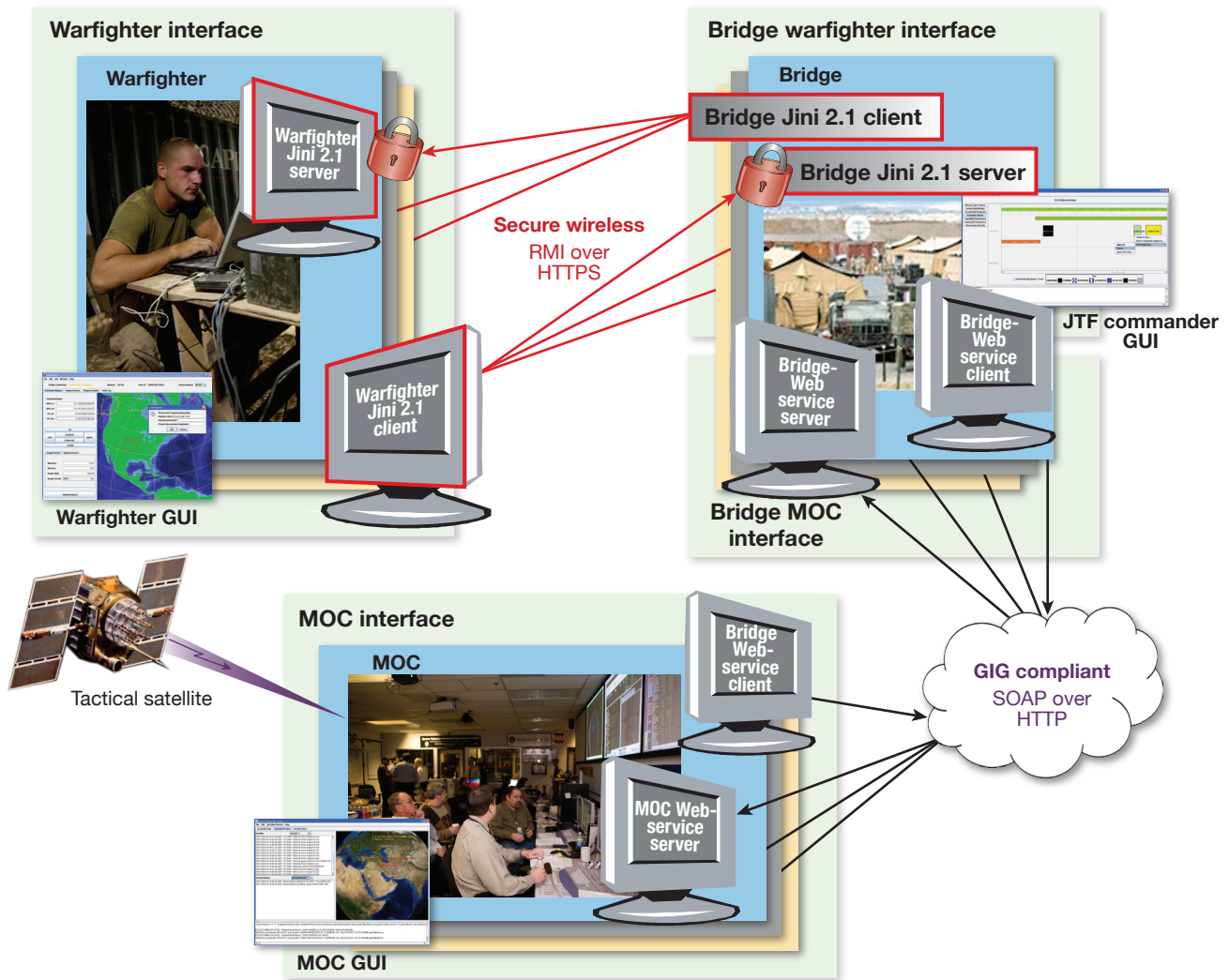


Figure 1. Notional TSCSA system architecture. HTTP, Hypertext Transfer Protocol; HTTPS, Hypertext Transfer Protocol Secure; RMI, remote method invocation; SOAP, Simple Object Access Protocol.

tactical spacecraft resource; the only conceptual change is the menu of capabilities that a specific payload provides. Also keep in mind that the capability dictionary exposes only safe operating modes that can do no harm to the payloads or spacecraft; this fact, and the easy-to-use ground system, is what keeps the training short and simple.

Our tactical spacecraft-commanding common ground system essentially boils down to the mobilization and processing of three primary Extensible Markup Language (XML) schema-defined data structures or message types: a capability dictionary, a target list, and an observation plan.

Capability Dictionary

Each MOC distributes capability dictionaries for the spacecraft and payloads it operates throughout the common ground system. Each common ground-system node ingests this dictionary and instantly knows the

capabilities provided by each payload. Capability dictionaries contain the following information:

- Spacecraft constraints
- Operational rules (i.e., which payloads can operate simultaneously)
- Commanding rules (i.e., duration and prerequisites for each instrument operation)
- Capability menu

Target List

Each warfighter uses the common ground system to add, modify, and retire targets from their local target list. The target list simply contains a listing of the areas of interest for a task force or warfighter. Each target in the list includes latitude, longitude, elevation above sea level, priority, and other planning constraints. Target lists may be exchanged via the TSCSA so that the MOC, or payload operations center, may act as a proxy for a warfighter by scheduling collections for their targets.

Observation Plan

The observation plan is just what it sounds like: a schedule of observations. In addition to target information and description, each observation in the plan has a status such as approved, pending, or request. So the plan provides a means for warfighters to express and exchange desired observations as well as a notification system for what is already approved for execution. The observations plan has the following information:

- Exclusion (nadir only or keep-out) zones, typically used for orbit trim maneuvers or momentum dumps
- Approved observations
- Requested observations with priorities
- Data downlink opportunities
- Predicted and actual (from telemetry) solid-state recorder levels
- Predicted roll profile
- References to spacecraft ephemeris, leap seconds, clock, and planetary kernels

Plan Generation (Observation Opportunity Search)

The common ground system provides a method for finding observation opportunities that satisfy adjustable constraints on roll angle, solar incidence angle, phase angle, etc. In some situations, warfighters or MOC personnel may want to run a search with a blank observation plan and loose constraints just to see what would be possible if there were no conflicts. Normal operations would entail finding opportunities that do not conflict with any previously approved observations and submitting an observation plan to the MOC or payload operations center.

Command Generation

Any ground system, at the MOC or at a warfighter, that has a capability dictionary, an approved plan, and spacecraft state kernels has all that is necessary to generate payload commands. If the approval is sufficient and the warfighter has the necessary communications gear, it would be possible for a warfighter to directly command a collection and retrieve the data. This has the possibility of enabling a single-pass commanding and collection scenario.

SERVICE-ORIENTED ARCHITECTURE

One of the major goals of the TSCSA research effort is to demonstrate the feasibility of developing a rapidly prototyped tactical spacecraft command system by using a framework of predefined common ground system interfaces and SOA support services. The framework must be highly generic and extraordinarily modular to successfully interface with a large heterogeneous set of existing spacecraft, ground, and “last-mile” communication systems. The SOA support services in the TSCSA were

built using Jini 2.1 to address some of these goals. Jini 2.1 is a specification for building SOA components that are autonomously discoverable across multiple networks, are platform independent, and are communications protocol agnostic. In addition, the services built using our Jini 2.1-based common ground system interfaces have runtime configurable security protocols and are capable of operating in secure wireless mobile *ad hoc* network (MANET) environments.

There are four primary SOA communication interfaces that flexibly connect warfighters coming in and out of communication range to communication bridges and tactical spacecraft MOCs over the GIG:

1. The warfighter interface
2. The communication bridge warfighter interface
3. The communication bridge MOC interface
4. The MOC interface

At a lower level these four interfaces are composed of four Jini 2.1-based client/server components operating in a secure wireless MANET environment and four World Wide Web Consortium (W3C)-defined Web-service client/server components operating inside the GIG:

- Warfighter Jini 2.1 client
- Warfighter Jini 2.1 server
- Bridge Jini 2.1 client
- Bridge Jini 2.1 server
- Bridge Web-service server
- Bridge Web-service client
- MOC Web-service server
- MOC Web-service client

The secure wireless MANET support provided by the Jini 2.1-based client/server components allows warfighters to come in and out of communication range and quickly pick up where they left off. This MANET capability also enhances the warfighter’s communication range by forwarding requests through other warfighters. By adhering to W3C-defined Web-service standards, the four Web-service components can simulate the operation of services operating inside the GIG.

Jini 2.1 SOA Components

The Jini 2.1-based bridge server component provides services to the Jini 2.1-based warfighter client component. These services include client registration, receiving and processing target requests, delivering status requests, and providing basic functionality such as client authentication and heartbeat monitoring.

The Jini 2.1-based warfighter server component primarily provides access to the warfighter’s imagery and imagery metadata repository for results delivery from a bridge client that wants to deliver the results of a requested target observation. Once a warfighter client has registered a request with a particular bridge, it does not need to be within range of that particular bridge to receive results. The warfighter server provides a stand-

alone results-delivery service that can be discovered and used by any bridge within range. Before accepting results from a bridge client, the bridge must first authenticate itself to the warfighter server for results delivery in the same way that warfighter clients must authenticate themselves to bridge servers to make a target request.

Web-Service SOA Components

The bridge Web-service client is used to subscribe to one or more available MOC Web-service servers. It is also used to pass along imagery and status requests from connected warfighters to the MOC Web services. Each bridge requires only a single Web-service subscription client to communicate with multiple MOC Web-service servers because all TSCSA image providers implement a common imagery provider Web-service interface. The bridge Web-service subscription client can communicate with different MOC Web services simply by selecting and unselecting the discovered providers in the prototype Joint Task Force (JTF) commander graphical user interface (GUI).

The MOC Web-service client listens for incoming warfighter imagery request results that were downlinked from a connected ground station and delivers the imagery to the bridge Web-service server that the warfighter happens to be connected to at the time the imagery is received. If the warfighter cannot be located at that time, the MOC Web-service client attempts to locate and deliver the imagery at regular intervals for a mission-defined timeout period.

PROTOTYPE APPLICATION

At the top level, the TSCSA prototype application is composed of three GUI components representing the warfighter, JTF commander, and MOC controller. These three user interface components are implementation-specific and were developed for the prototype application to demonstrate the underlying TSCSA framework. The three user interface components work together to process and transport capability dictionaries, target lists, and observation plans by using the processing algorithms and communication interfaces of the TSCSA.

Warfighter GUI

The warfighter GUI, shown in Fig. 2, provides an intuitive mechanism for making target requests and viewing delivered tactical spacecraft imagery. The warfighter GUI essentially wraps the core TSCSA war-

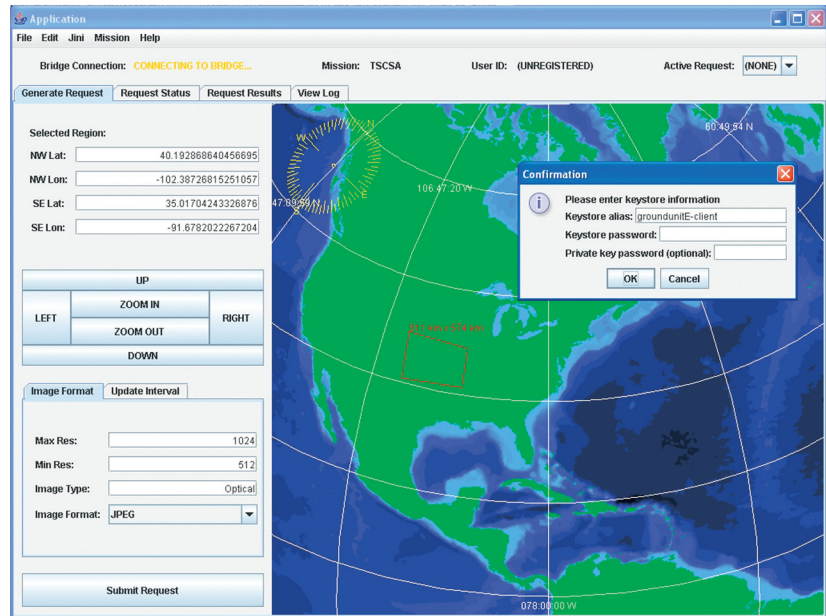


Figure 2. Warfighter GUI (bridge discovery and secure login).

fighter functionality in an easy-to-learn and easy-to-use GUI. The GUI menu provides access to the Jini infrastructure needed to autonomously discover and interact with other wireless Jini components in the network.

Users can generate a new target request by using the interactive 3-D display to zoom in on any geographic location in the world. Then by dragging within the view window (Fig. 3) the user can obtain the latitude and longitude of an area of interest. Once a region has been selected, the user simply specifies an image format and update interval and submits the request to any bridge that happens to be within range.

Once the user has submitted a target request, the status of the request can be checked at any time. The GUI provides a status history tab that allows the user to quickly ascertain the likelihood that a particular request will be fulfilled and to find out the estimated time of arrival of the associated imagery results. If the user decides the request is no longer necessary, it can be retracted up to a certain critical amount of time before the associated command load.

After a target request has been approved and the associated command load has been successfully scheduled, uploaded, and executed on board the spacecraft, the resulting observations are downlinked and delivered to the warfighter. The warfighter GUI allows the user to examine the results of a particular target request in detail by displaying imagery metadata and allowing the user to zoom in on and drag-scroll the resulting images.

JTF Commander GUI

The JTF commander GUI (Fig. 4) displays all the transactions that take place between connected warfighters and selected MOCs. It provides the JTF

commander with an easy-to-use interface for choosing which of the discovered MOCs should be made available to service warfighter target requests.

The JTF commander GUI, shown in Fig. 5, shows the warfighters that are currently connected to the bridge and displays useful statistics for all the warfighters involved in a mission. If more information is desired, the JTF commander can use the GUI to drill down and examine the details of an individual target request and/or monitor the status history of a target request in real time as it is updated by the MOC Web services with which the JTF commander has registered.

A capability dictionaries tab in the GUI allows the JTF commander to view the capabilities of each asset available for commanding at the various registered MOCs, and a command sequences tab allows the JTF commander to examine the details of the command sequences generated for a particular collection event. Figure 6 shows a menu of capabilities provided by the satellite asset.

For the prototype application, a real, working capabilities dictionary was built for the APL-designed Compact Reconnaissance Imaging Spectrometer for Mars (CRISM) instrument aboard the NASA Mars Reconnaissance Orbiter (MRO) spacecraft. After registering with the prototype MOC interface implementation (see the MOC GUI section), the CRISM instrument's capabilities dictionary is automatically delivered and loaded into the JTF commander GUI. The real-world instrument capability definitions in this dictionary can be used subsequently to generate real MRO observation plans and actual CRISM command sequences in response to warfighter target requests approved by the JTF commander (the only caveat being that the latitude and longitude in the warfighter target request is interpreted as a latitude and longitude on the surface of Mars in the observation plan generator).

Figure 7 shows a schedule viewer in the GUI that displays a real-time updated view of the latest orbit, track,

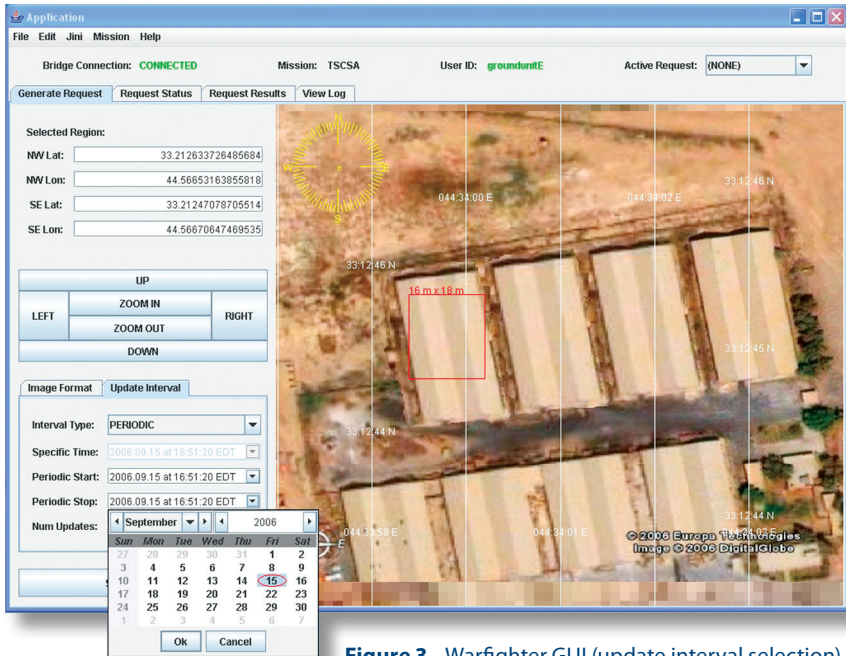


Figure 3. Warfighter GUI (update interval selection).

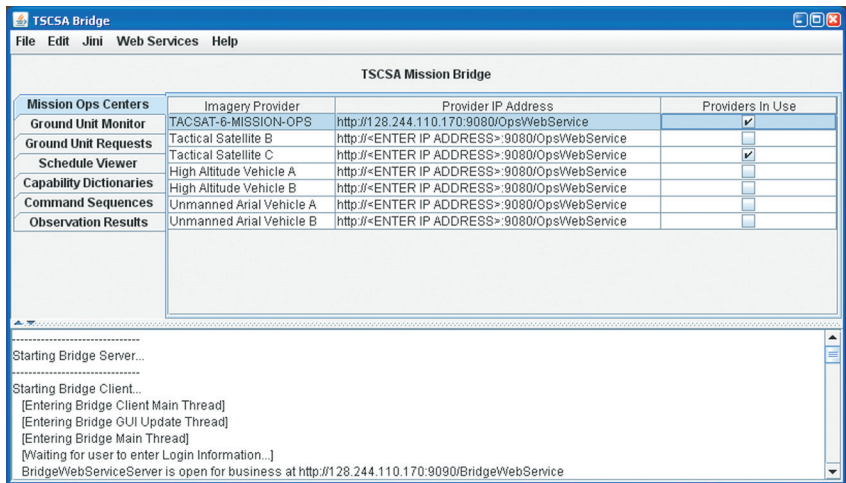


Figure 4. JTF commander GUI imagery provider selection tab.

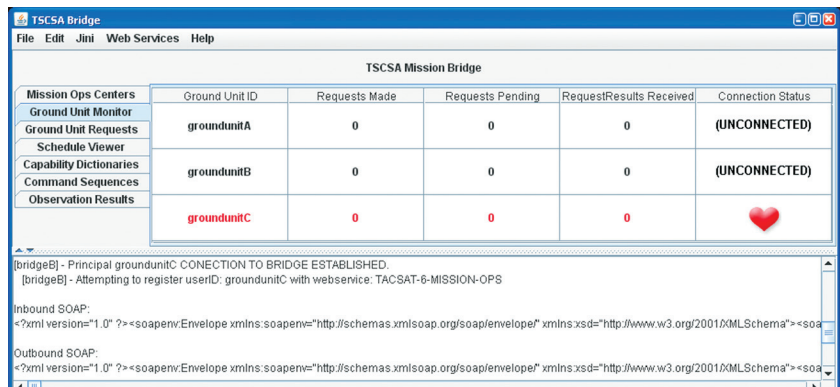


Figure 5. JTF commander GUI warfighter heartbeat monitor tab.

and event schedules of each asset being commanded by the coordinating MOC. When a warfighter makes a target request, the bridge takes into account the capabil-

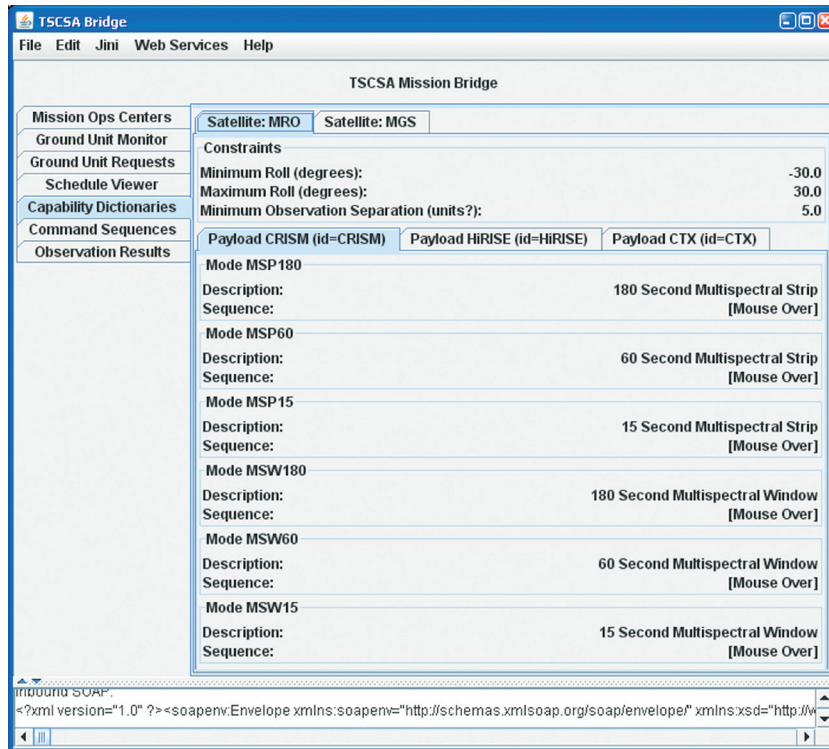


Figure 6. JTF commander GUI capability dictionaries view.

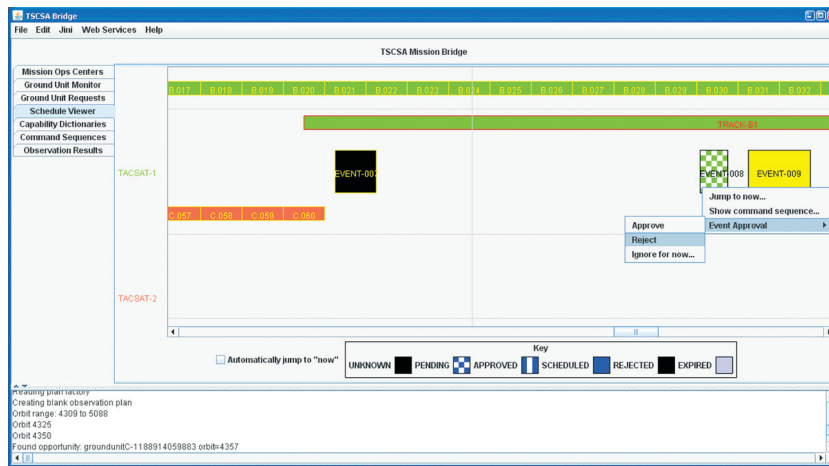


Figure 7. JTF commander GUI schedule viewer.

ities dictionary associated with each asset and considers the latest orbit, track, and event schedules received from the MOCs and attempts to generate an observation plan for the asset that fulfills the target request.

If the observation plan generator is able to generate the set of events necessary to fulfill a request, then the schedule viewer is updated with tentative uplink, downlink, and collection events. The JTF commander then manually accepts or rejects the new tentative event schedule by right-clicking on a tentative collection event and accepting or rejecting it. If the JTF commander chooses to reject the collection event, then all the associated

tentative uplink and downlink events are removed from the schedule and the warfighter is notified that his request was rejected. If the JTF commander chooses to accept the collection event, the actual command sequences necessary to carry out the collection event are generated and the new events schedule and necessary command sequences are immediately delivered to the appropriate MOC(s) and/or ground station(s) for upload to the spacecraft.

The final tab in the JTF commander GUI displays details of any collection event result that has been delivered to the bridge from a connected MOC Web-service imagery provider (Fig. 8). The same metadata information available to the warfighter is also available to the JTF commander in this display. The results view also provides information on the delivery status of a collection event result and from where the result came.

MOC GUI

The MOC GUI serves as the TSCSA imagery provider interface implementation as well as a spacecraft monitor and simulator and ground station monitor and simulator in the prototype. An operations view tab provides an interactive 3-D NASA World Wind-based spacecraft to ground station monitor. The MOC controller can use the mouse to navigate the globe and zoom in on any geographic location while relevant orbit, track, and event visualizations are updated in real time.

The MOC GUI also has a collections result view and event scheduler similar to the ones on the bridge, as shown in Fig. 9. Changes in asset schedules are automatically kept in sync as incoming schedule updates are received from registered bridges, and the result view is automatically updated as collection event results are downlinked from a spacecraft.

Timing Web-Service GUI

For demonstration purposes the TSCSA prototype also contains a timing Web-service GUI (Fig. 10) for setting the current time used to synchronize TSCSA

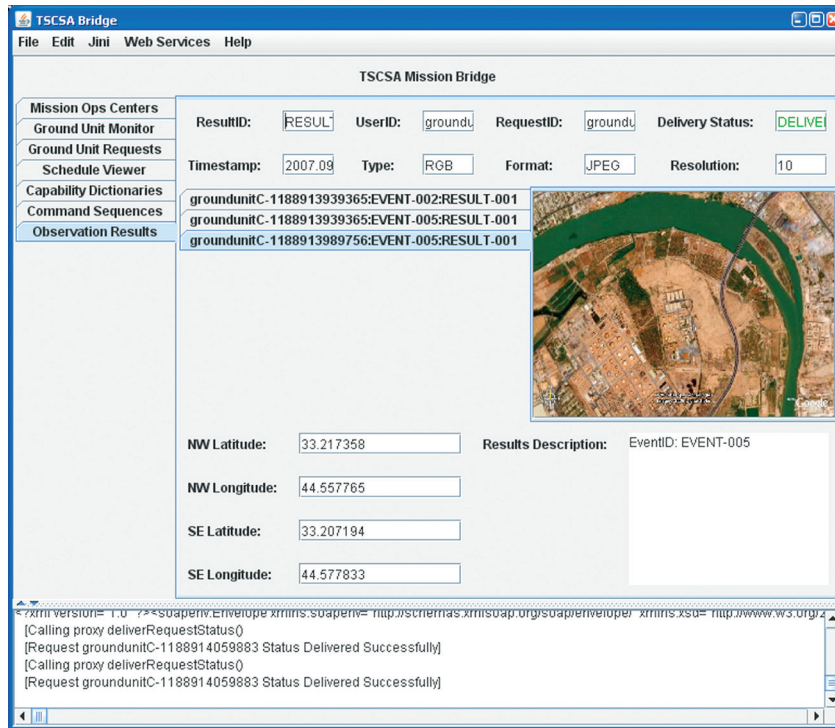


Figure 8. JTF commander GUI result viewer.

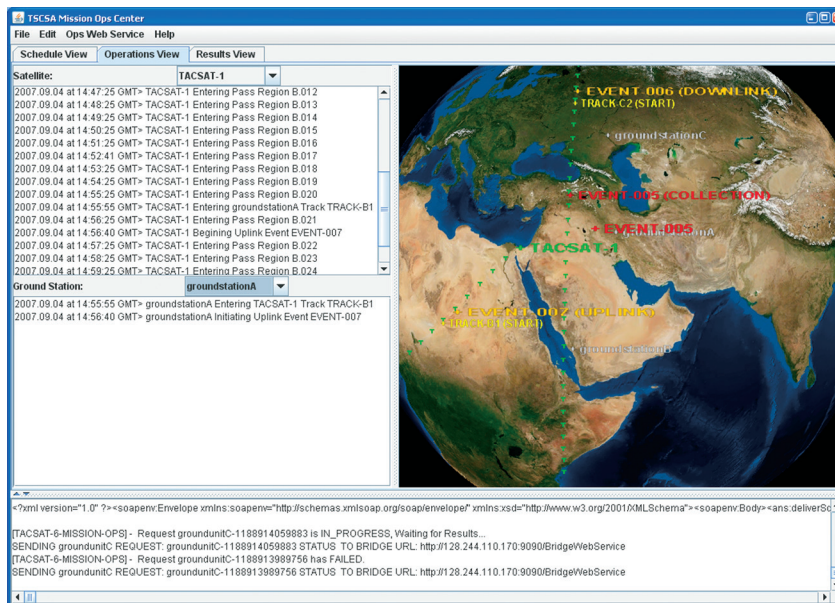


Figure 9. MOC GUI operations view.

components with each other. The timing Web service also allows the TSCSA developer to adjust the timescale to speed up or slow down the demonstration.

FUTURE WORK

The design and configuration of this architecture was born of experience with civilian space programs. There-

fore, the infrastructure needs to be tested and vetted in a military scenario with a realistic operational tempo. One of the tactical satellite programs would make a good candidate for experimentation, but this architecture may perform equally well in a traditional reconnaissance operation. One possible way forward would be to work our architecture and process side by side with existing military space processes to identify needed enhancements to tools and assess the ability to keep up with ongoing operations.

The Joint Tactical Radio System (JTRS), currently under development by the U.S. military, is a next-generation voice-and-data Software Defined Radio system built with the open source Software Communications Architecture framework. This framework represents a major paradigm shift from traditional radio systems that rely on hardware to define their communication waveforms. One exciting aspect of JTRS is that it will enable integrated encryption for IP-based waveforms in a wideband networking environment. These capabilities will in turn enable the creation of secure robust MANETs that are capable of operating reliably over large distances. The current TSCSA prototype operates within a secure wireless MANET environment built using security-agnostic and communications-protocol-agnostic Jini 2.1 SOA components, but the underlying communications hardware in the prototype consists of nothing more than common 802.11-based COTS products (the same products used to build residential wireless networks).

The 802.11-based COTS products are only capable of operating over short distances and are not an acceptable solution in most real-world warfighter scenarios. In order to demonstrate the usefulness of the TSCSA framework to future war-fighter-oriented tactical satellite applications, a JTRS-based TSCSA prototype could be built and field tested in a nonlaboratory environment.

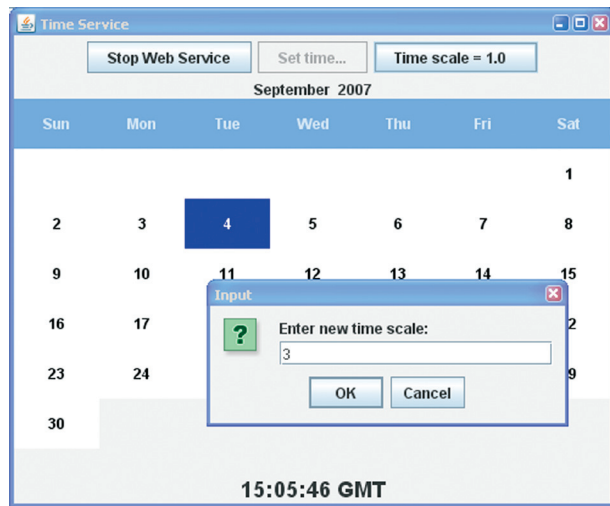


Figure 10. Timing Web-service GUI.

SUMMARY

This research built a flexible ground system and commanding capability into a Web-services architecture that shows promise for placing fully capable payload operations centers into the hands of distributed warfighters with minimal training. The Web architecture

makes it possible to select desired services from a menu of services and providers and track requests through the scheduling, execution, and downlink phases. Selection and tracking of services represents a radical paradigm shift compared with existing lengthy and opaque processes. The nature of Web services makes deployment and training for applications and upgrades, even to distant locations, trivial. Most moderately Web-savvy people have used Web pages and forms similar to the ones that make up this architecture. In list form, the accomplishments of the TSCSA team are:

- Architecture and design of a Web-services framework
- Development of tactical satellite planning and commanding software components
- Deployment of a SOA to a mobile computing laboratory
- Laboratory testing

The purely technical aspects of this architecture have been designed, coded, and tested for capability, accuracy, and robustness. By those metrics the research has turned out extremely well. The practical considerations of clearance, colocation, and access to sensitive information make the prospects of performing a realistic integration and testing protocol more challenging.

The Authors



Puck-Fai Yan



Matthew J. Potter



James A. McGovern

Puck-Fai Yan is a member of the APL Principal Professional Staff in the National Security Technology Department. He is currently Supervisor of the Information Systems Section of the Systems Concepts and Analysis Group. He holds M.S. and B.S. degrees in electrical engineering from The Johns Hopkins University and the University of Pittsburgh, respectively. Mr. Yan has more than 20 years of experience in software and systems engineering in undersea warfare, space, and homeland protection information systems development. He held technical and management positions in both the National

Security Technology Department and the Space Department as assistant group and group supervisor. He was the Principal Investigator of the National Security Space Business Area IR&D project on the TSCSA and conducted research in support of the Operationally Responsive Space concept. **Matthew J. Potter** is an Associate Professional Staff member of the APL Space Department Science Applications Group. He earned B.S. and M.S. degrees in computer engineering and minors in mathematics and artificial intelligence from North Carolina State University. The topic of his master's thesis was the proposal of a novel artificial neural network learning algorithm. Since joining APL in 2003, he has worked on a variety of projects ranging from common ground systems development for the New Horizons, STEREO (Solar Terrestrial Relations Observatory), and MESSENGER (Mercury Surface, Space Environment, Geochemistry, and Ranging) missions to rapid prototype development for the Satellite-As-Sensor IR&D, the Virtual Ionosphere Thermosphere Mesosphere Observatory, and the Chipotle IV Base Station. **James A. McGovern** is a member of the APL Senior Professional Staff in the Space Department. He received a B.S. in aerospace engineering from the University of Maryland and an M.S. in electrical and computer engineering and an M.S. in applied physics from The Johns Hopkins University. Mr. McGovern's work focuses on science operations for the CRISM instrument on board the MRO and the automated landing system for the Moon. For further information on the work reported here, contact Puck-Fai Yan. His e-mail address is puck.yan@jhuapl.edu.

The Johns Hopkins APL Technical Digest can be accessed electronically at www.jhuapl.edu/techdigest.