Onboard Autonomous Planning System

Ella Herz1, Josh Neel2, Marc Simpson3, and Doug George4

Orbit Logic, Greenbelt, Maryland, 20770

Tim Esposito5
Emergent Space, Greenbelt, Maryland 20770

and

Ken Center6
PnP Innovations, Albuquerque, NM 87109

Under a Phase II Small Business Innovation Research (SBIR) contract with the US Air Force Research Laboratory (AFRL) at Kirtland Air Force Base, the team of Orbit Logic, PnP Innovations, and Emergent Space are developing an Autonomous Planning System (APS) that can run onboard a satellite in flight software and respond to onboard and external events to meet the planning/scheduling requirements of a variety of missions. The modular architecture allows planning systems to be assembled from individual planning components and quickly configured (and reconfigured as necessary) to meet initial and dynamic mission goals.

Nomenclature

AFRL = US Air Force Research Laboratory
AMM = Autonomous Mission Manager
APS = Autonomous Planning System
ASPIRE = Adaptive, Scalable, Portable Infrastructure for Responsive Engineering
GEMS = Ground Enterprise Management System
SBIR = Small Business Innovation Research
MAPA = Master Autonomous Planning Agent
MCV = Mission Configurable Variable
SAPA = Specialized Autonomous Planning Agent
SBIR = Small Business Innovation Research

I. Introduction

Spacecraft planning and scheduling is currently performed on the ground, with resulting plans uplinked to the spacecraft for execution. This process has inherent time delays that can cause a loss of critical mission opportunities or cause the loss of the spacecraft.

If a potential opportunity is detected by one or more spacecraft it is much faster to let the spacecraft plan their own response than it would be to generate response plans on the ground and transmit those plans to each spacecraft. When opportunities are detected onboard, the spacecraft can react immediately without the need for communications coverage or the need for a ground planning response. Faster response times lead to better responses and enhanced mission success.

1 Chief Operating Officer, Orbit Logic, 7852 Walker Drive, Greenbelt, MD.
2 Principal Software Engineer, Orbit Logic, 7852 Walker Drive, Greenbelt, MD.
3 Senior Systems Engineer, Orbit Logic, 7852 Walker Drive, Greenbelt, MD.
4 Vice President, Orbit Logic, 7852 Walker Drive, Greenbelt, MD.
5 Director of Mission Operations Technologies, 6411 Ivy Ln, Greenbelt, MD
6 Director of Advanced Software Technologies, 3921 Academy Pkwy N NE, Albuquerque, NM.
Under a Phase II Small Business Innovation Research (SBIR) contract with the US Air Force Research Laboratory (AFRL) at Kirtland Air Force Base, the team of Orbit Logic, PnP Innovations, and Emergent Space are developing an Autonomous Planning System (APS) that can run onboard a satellite in flight software and respond to onboard and external events to meet the planning/scheduling requirements of a variety of missions. The modular architecture allows planning systems to be assembled from individual planning components and quickly configured (and reconfigured as necessary) to meet initial and dynamic mission goals.

The onboard Autonomous Planning Agent (APS) operates using a rolling timeline, constantly adding or modifying the existing spacecraft Command Queue as new information is received in the form of events. The APS planning timeline is configurable, but is envisioned as a very short timeframe (on the order of a few minutes to a few hours at most). The APS can be considered a “just-in-time” planner. The APS is generally not used to generate a long-term plan, such as a day’s worth of activities. The APS uses specialized planning agents to determine what to do over the next few minutes or hours based on the latest system state and in response to triggering events.

APS includes Specialized Autonomous Planning Agents (SAPAs) that address specific planning needs (recorder management, ground target imaging, collision avoidance, etc.) and can support SAPA-specific algorithms for each planning domain. This approach is unique and contrasts with the current state-of-the-art for planning systems which generally try to apply a single algorithm type (often state-based and rules-based for flexibility) to multiple planning domains, often very inefficiently and beyond the processing capability of onboard resources.

Plans generated by multiple SAPAs are integrated by a Master Autonomous Planning Agent (MAPA) that deconflicts global resources and forwards the final plan to the onboard task executive for implementation.

Giving satellites the ability to make autonomous planning decisions allows satellites to respond much more quickly to capture opportunities that might otherwise be missed. The MAPA/SAPA architecture for onboard planning supports flexibility to plan for different kinds of opportunities, keeps the system modular and efficient enough to be used in an onboard environment, and makes the system extensible to almost any satellite planning domain.

The MAPA/SAPA architecture and short planning timelines lend themselves to coordinated constellation planning because the individual components do not care where the event messages originate (on the same satellite, a different satellite, or the ground), and planning can be performed and re-performed as different systems react to the environment as understood from event messages on the bus. As more spacecraft need to coordinate activities to reach specific goals as a whole, a configurable and adaptable planning architecture becomes more critical.

The APS architecture could be applied to reduce the cost, schedule, and risk of implementing planning systems on various platforms, while at the same time making the resulting planning systems more agile to respond to dynamic mission goals and more efficient with the use of processing resources.

II. Prior State of the Art of Satellite Mission Planning

Most satellite mission planning is currently performed on the ground with resulting schedules transmitted to satellites (via command load) during periods of communications availability (generally during periodic ground station contacts). For some of the satellite missions that have implemented onboard autonomy, such as Air Force’s TacSat-3 [Reilly 2006] and NASA’s EO-1 [Chien et al. 2010] missions, the solution was very mission-specific and could not be considered a modular architecture easily adaptable to new missions, and/or used state-based and rules-based planning that cannot scale to meet complex planning needs within the constraints of onboard resources.

The ground planning approach inserts significant delays in responding to new information and new opportunities. These delays, which can range from several minutes to several hours, can mean the loss of opportunities for the collection of critical intelligence data, the degradation of the satellite due to a slow response to a system failure, or even the loss of the satellite (such as the case of a collision avoidance).

Mission planning can be a complex process requiring significant processing power and memory, but it does not have to be. Many state-of-the-art planning systems are state-based and rule-based because this provides significant flexibility, but that flexibility comes at the price of processing resource requirements which are not generally available onboard a satellite. By breaking the planning system into specialized, yet coordinated components, the entire system is made more efficient and flexible. Specialized planning components (like the SAPAs proposed in our Phase II effort) support the use of the most efficient domain-specific algorithm and also allow planning systems to be more easily developed, adapted, and assembled to meet mission-specific goals.

By using a rolling planning timeline, reducing the planning window timeframe and using specialized and efficient planning agent components, useful mission planning can be performed by available satellite processing resources onboard.
III. APS Concept of Operations

This section summarizes the current standard mission planning concept of operations and its major deficiency and describes a new concept of operations that includes the proposed onboard APS and how it addresses the deficiencies of the current concept.

A. Concept of Operations Background

Current satellite mission planning is performed on the ground. Planning periods are generally constrained to timelines bound by contact opportunities where command loads to execute those plans can be uplinked. Due to this limitation, a satellite command load can only address specific events if the mission planning group knows about the event prior to the start of the planning process. This concept of operations is not equipped to respond to opportunities that are detected between commanding opportunities, which can be several minutes or even hours apart. Figure 1 shows a traditional mission planning timeline.

A planning agent located onboard the spacecraft eliminates this major deficiency of the existing mission planning concept of operations by making appropriate changes to the existing spacecraft command load (or generating a new load from scratch) in response to events in near-real-time. The onboard planning agent can generate and/or maintain a plan that responds to evolving conditions between ground uplink opportunities.

Migration of mission planning activities to an autonomous flight software agent will allow future missions to implement true real-time opportunistic target collections, and other unrealized capabilities enabled by onboard planning.

![Figure 1. Traditional Ground-Based Mission Planning Timeline.](image)

A. APS Operational Overview

APS monitors messages on the spacecraft message bus and generates new actions for the onboard task execution engine in response to events detailed in these messages. Messages may contain spacecraft state information and/or external event information detected by onboard sensors, external space systems, or by the ground.

To understand the activities of the APS in more detail, Figure 2 depicts a use case diagram where an action is generated as a response to a detected event.
1. The APS is notified of an external event via a separate on-board Event Detection Agent. This new event is placed in the Event Queue.
2. The Event Filter pulls the events from the Event Queue and applies Planning Window and time constraint criteria. All of the events that do not pass the filter are sent back to the Event Queue.
3. Events that meet the Planning Window criteria are sent to the Decision Algorithms.
4. The Decision Algorithm pulls the state information for the start of the Planning Window from the Action Queue and uses that information to enforce spacecraft constraints in the plan under consideration.
5. Once a valid solution has been found and a plan has been verified it is transmitted to the Action Queue where it is executed at the planned time.

B. APS Planning Timeline

The onboard Autonomous Planning Agent (APS) operates using a rolling timeline, constantly adding or modifying the existing spacecraft Command Queue as new information is received in the form of events. The APS planning timeline is configurable, but is envisioned as a very short timeframe (on the order of a few minutes to a few hours at most). The APS can be considered a “just-in-time” planner. The APS is generally not used to generate a long-term plan, such as a day’s worth of activities. The APS uses specialized planning agents to determine what to do over the next few minutes or hours based on the latest system state and in response to triggering events.

APS specialized planning agents generate plans (schedules of activities) for their specialties (such as imaging ground targets or adjusting the orbit to avoid a collision) and output those plans to a master planning agent that deconflicts the use of common resources (like spacecraft attitude, power consumption, recorder space, etc.) and outputs messages to update the current Command Queue.

Planning onboard is unbound from command uplink opportunities and uses system state information and events available onboard to drive its planning. The APS planning timeline is shown in the figure below.
The APS planning timeline is set by the following Mission Configurable Variables (MCV):

- Planning Window (PW) duration: Duration of time for the APS to consider for a single plan. This will be dictated by mission specifics (i.e. orbit, sensor, maneuverability, etc.) Planning Window start time is set to current time plus Planning Time (PT).

- Planning Time (PT) duration: Amount of time the APS is allowed to prepare a plan prior to Planning Window start time. This is determined by a formula \((PW/x) + d\) where \(PW\) is the Planning Window duration, \(x\) is an MCV that ensures that Planning Time is less than the Planning Window duration, and \(d\) is Action Transmit Time. Action Transmit Time is a configurable buffer that represents the time between the transmit of a new action from the APS and insertion of that action into the Command Queue prior to execution.

C. APS Operational Modes

The role of the mission planner on the ground (whether a person or an autonomous system) will evolve as the mission planning activities shift to an onboard autonomous agent. The role of the ground operator will depend on the autonomy mode selected and the extent of the onboard autonomous planning system agents. The following sections describe the autonomy modes for the APS, and the role of the ground in each. The ground role may be filled by operator(s) or autonomous systems. Note that the autonomy mode may be selected for each individual specialized planning agent separately, or for the entire APS as a whole. So a satellite may be by using APS to operate autonomously for certain functions (such as collision avoidance and ground imaging) but rely on ground planning for other functions like contact scheduling and recorder management.

The Autonomous Planning System and its individual components can be configured for fully autonomous operation, semi-autonomous operation, or it can be disabled so the spacecraft depends fully on command loads generated on the ground. There is also an interrupt sub-mode while operating in either full- or semi-autonomous mode.

Table 1. APS Operational Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Operational Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Full-Autonomous</strong></td>
<td>Mission planners do not need to generate a command load for uplink to the spacecraft. The APS will take input from the event detection agents and the other interfaces to create the actions that will be executed by the onboard task execution engine.</td>
</tr>
<tr>
<td><strong>Semi-Autonomous</strong></td>
<td>The design of the APS allows it to work in parallel with, and augment, traditional ground based planning. The APS would operate using the same process as full-autonomous, but the APS would only add actions to the Action Queue when it determines that the external event is above a configured priority threshold and that no actions in the Action Queue of a higher priority are overwritten. In this mode the APS remains dormant until an action with a priority above the threshold is placed in the Event Queue.</td>
</tr>
</tbody>
</table>
It is likely that this mode will be the primary mode used for early APS deployments. It allows
the ground to continue to plan traditionally, but enables autonomous spacecraft actions in
response to high priority opportunities.

| Disabled | The APS does not perform any planning. All planning is performed on the ground with
command loads uplinked from the ground controlling all satellite activities. |

External events that are critical to the satellite mission trigger an Interrupt Mode during full-autonomous or
semi-autonomous APS operations. Events that trigger Interrupt Mode are set by MCV. Once these events are
detected, the existing onboard planning timeline is interrupted and a new plan and associated actions are generated.
Interrupt mode would be useful to schedule an orbital adjustment to avoid an impending collision.

Even when operating in fully-autonomous mode, human input and oversight will still be required by the APS.
Mission Planners on the ground can still influence the activity of the spacecraft by:
- Generating and uplinking event messages (as needed) to trigger desired APS activity
- Monitoring APS activity to ensure mission objectives are being fulfilled
- Adjusting APS MCVs as needed to meet mission objectives
- Selecting the desired APS mode based on the mission environment and APS design
- Generating and uplinking plans during semi-autonomous and disabled mode periods
- Assessing APS responses to events and situations using an APS ground simulator

IV. Specific APS Use Cases

The APS architecture can support an almost unlimited number of use cases for autonomous planning given the
specialized planning components. The following list briefly describes the relevant use cases that are the focus of our
Phase II SBIR effort and the role of the APS in each.

A. Onboard cueing of opportunistic high-resolution imagery collection

A wide area sensor onboard the satellite detects an unexpected ground-based event prior to the location entering
the field of view of a broadside synthetic aperture radar (SAR) sensor. The APS determines the optimal beam for
the collection opportunity from the input ground point event coordinates and adjusts the current collection schedule
to include the high value event first.

B. Collision avoidance maneuver

APS is alerted to an impending collision with a piece of “space junk”. The current APS planning cycle is
interrupted, APS computes and schedules an avoidance maneuver, and sends events to the onboard execution agent
which commands the propulsion and attitude subsystems to execute the maneuver. The collision is avoided and the
spacecraft survives. APS auto-adjusts the pending mission schedule to compensate for the change in orbit. It is
worth noting that the current collision avoidance approach is based on information in space catalogs that may not be
complete, must be done far in advance of possible collision events, and relies on very computationally intensive
analysis that most operations centers won’t be able to perform. Using APS with the appropriate onboard sensors
will enable real-time collision avoidance that was not previously available.

C. Imaging of space object

A space object has been detected and located. The APS onboard an imaging satellite sees this event on the
spacecraft message bus and autonomously plans a series of spacecraft actions (including attitude maneuver, sensor
configuration/activation, and image storage/downlink) to collect and return an image of the space object when it is
in view on the next orbit. The APS adjusts the current routine imagery collection schedule to insert this high priority
target.

V. APS Architecture

D. Flight Software Architecture

The APS is a component that will interface with other components in a flight software system as shown in Figure
4. Being a flight software component, the APS must be flexible in terms of supporting multiple operating systems,
hardware platforms, and flight software architectures. Supported flight operating systems include RTEMS and
VxWorks. Supported hardware platforms include PowerPC, SPARC, and x86. ASPIRE is the flight software
messaging framework being used by the Autonomous Mission Manager (AMM) out of AFRL Kirtland. APS sub-components communicate with each other and with external flight software components via middleware such as ASPIRE messages as shown in the figure below. This architecture lends itself to extensibility. As many (or as few) onboard specialized planning agents can be included in the flight software as desired for as many planning functions as are needed for a particular mission. Additional planning agents can be added later to expand onboard autonomy, if desired.

Figure 4. APS Layered Architecture with ASPIRE.

PnP Innovations facilitated the rapid accommodation of the APS into the AMM framework by setting up a development environment to enable the team to concentrate on their migration and testing of autonomy modules. This will consist of tailoring a simulation environment to host a candidate satellite configured with the support devices and mission sensors necessary to accomplish the mission operational goals necessary to showcase the autonomous on-orbit planning offered by the team. This would also include defining and parameterizing specific orbits, communications ground sites, orbital conditions, and other satellites that are of consequence to performing high-fidelity mission simulation.

Figure 5. Simulation and Flight Software Development Environment.
Figure 5 illustrates the key features of the development environment that PnPI stood up in support for APS. The simulation framework hosts satellite support device models that present data interfaces on the ASPIRE operational domain. Domains in ASPIRE allow the scope of visibility between components to be controlled on the same physical network. This allows multiple satellites to be hosted in the simulation and their devices and applications to interact without interference. This is heavily utilized for exchanging state data between different simulation frameworks, a capability that will be employed on an as-needed basis to leverage COTS infrastructure that speeds our way toward gap-filling capability toward a robust mission simulation. APS Agents developed and integrated with this framework are wrapped within an ASPIRE API and allowing them to freely interact with other classes of flight system software, including:

- Onboard services such as data storage interactions, time management, and math libraries.
- Mission services such as slew and celestial body keepout planning support services, orbit propagators, etc.
- Interfaces to satellite subsystem control (Attitude Determination and Control, Power Management, and Communications.

In the case of the latter, the communication subsystem controller interacts through a radio model to allow accurate simulation of command and telemetry exchanges between the ground station and the satellite. The ground station consists of an ASPIRE Data Browser providing the same capabilities as the “Development and Test Support” Browser residing on the simulation workstation.

E. Autonomous Planning System Architecture

The core components of the APS include the Master Autonomous Planning Agent (MAPA) and a collection of multiple Specialized Autonomous Planning Agents (SAPAs) that meet the needs of a particular mission. The layers and components for an APS ASPIRE deployment are shown in Figure 6. ASPIRE contains the reusable services for all spacecraft missions and is the runtime environment for the flight software. In the APS layer, there is a collection of planning agents to be triggered by external events and to perform the planning process. To interface with other services, the APS ingests and reads ASPIRE messages. ASPIRE provides the middleware. Tables are used to manage and control the configuration of the APS through Mission Configurable Variables (MCVs). The final product of the APS is a plan, which is executed onboard the spacecraft and also downlinked to the ground.

The APS is implemented as a suite of ASPIRE applications. The MAPA and each SAPA each exist as independent ASPIRE applications, allowing mission-specific configurations of APS components to be defined. The source code for each planning agent is compiled into a single library file that can be loaded by the ASPIRE runtime environment. Tables for each planning agent are also compiled and deployed to the same location as the ASPIRE application shared object files.

F. Master Autonomous Planning Agent (MAPA)
In a planning and scheduling system a central component needs to gather plans from multiple agents, contain an agenda with priorities and rules for resolving conflicts, check for global constraint violations, perform conflict resolution, and interface with the command execution engine. The MAPA is the component that performs these functions for the onboard APS architecture.

The MAPA receives schedule input from multiple SAPAs (via ASPIRE messages), generates an integrated schedule of activities, verifies and de-conflicts that schedule against common resource constraints, and produces a final activity schedule that is implemented by sending commands to the onboard task execution engine.

The MAPA process flow, interfaces, and data structures all conform to AMM standards. The MAPA has interfaces to accept ASPIRE message inputs from multiple SAPAs and standard output interfaces to send ASPIRE messages to the task execution engine and back to the SAPAs. The MAPA also contains a set of mission-specific rules for constraint verification and deconfliction. The MAPA process flow is shown in Figure 7.

![Figure 7. MAPA Process Flow.](image)

The MAPA constraint verification and de-confliction algorithms are being adapted for flight software from Orbit Logic’s existing algorithms. The MAPA itself, while a critical component of the onboard APS, has relatively lightweight planning and scheduling responsibilities that consist of integrating the plans from multiple SAPAs. The complex planning is performed by the SAPAs themselves. The MAPA is used to verify that global resource
constraints are not violated by the integrated multi-SAPA plan. When there is a conflict or constraint violation, the MAPA uses a simple, configurable priority system to remove independent SAPA-defined activities until the conflict is resolved.

G. Specialized Autonomous Planning Agent (SAPA)

The APS for a particular mission may include multiple SAPAs such as a SAR Ground Imaging SAPA and a Collision Avoidance SAPA. The SAPA components of the proposed APS architecture are the true strength of the APS, and an acknowledgement that different kinds of operations require different kinds of planning routines and algorithms. Planning and scheduling is not a one-size-fits-all kind of function. Each SAPA can expertly perform planning for a different kind of operation (imaging vs. orbit maneuvering vs. downlink planning), and the MAPA is used to generate the integrated, deconflicted multi-SAPA schedule for execution.

The APS architecture is scalable to support multiple specialized planning components (SAPAs). As part of the Phase II SBIR effort, Orbit Logic is developing three SAPAs:

- SAR Ground Imaging SAPA
- Collision Avoidance SAPA
- Space Imaging SAPA

Some details of each of these SAPAs are further discussed in subsequent proposal sections.

The basic SAPA process flows (common to all SAPAs) are shown in Figures 8 and 9. Figure 10 describes a loopback process where actions scheduled by a SAPA are subsequently reverted (removed from the plan) by the MAPA then returned to the Event Queue for consideration in a future planning window. An optional Telemetry Verification process (occurring outside of the APS) is also depicted where events are returned to the Event Queue when the associated actions have been published with success criteria and actual spacecraft telemetry values have had out of limits violations indicating that the MAPA-published actions did not execute as planned.
Figure 8. Main SAPA Process Flow.

Figure 9. SAPA Event Filter Process Flow
H. SAR Ground Imaging SAPA component

The objective of this SAPA is to plan a schedule of ground image collections using a broadside SAR sensor on the satellite. The collection schedule is driven by the latest satellite ephemeris data available onboard, the latest set of targets of interest (provided by ASPIRE event messages from onboard or external sources like the ground or other satellites), and sensor attributes.

The SAR Ground Imaging SAPA has specialized internal logic and algorithms to compute access times from configurable SAR beams to targets of interest during the current rolling planning period. Using a figure-of-merit the most valuable constraint-conforming beam with target access will be selected and on/off times for the beam will be computed within configurable imaging buffers. For situations where multiple targets have overlapping or conflicting beam access, the same figure-of-merit is used to select the most valuable target acquisition. At the end of the process, the SAPA outputs a deconflicted high-fidelity target acquisition schedule for the current planning period as an ASPIRE message for the MAPA. The SAPA retains SAR target fulfillment status for all identified targets for use during future planning windows, and updates that fulfillment status based on ASPIRE telemetry messages from the MAPA or other external flight software components.

I. Collision Avoidance SAPA Component

The objective of this high priority SAPA is to plan orbit adjust maneuvers when required to avoid collisions with space objects. It monitors the message bus for event messages (that could originate from onboard sensors, other satellites, or the ground), compute collision likelihood, and quickly plan orbit adjust maneuvers when collision likelihood is above a configurable threshold. Orbit adjust maneuver computations will be based on current spacecraft attitude, attitude control characteristics (if required to point thrusters) and onboard thruster attributes.

Collision likelihood determination consists of propagating orbits and computing closest approach between the satellite and space junk. When the computed closest approach distance is less than a configurable value, the SAPA will initiate its avoidance maneuver logic. Orbit adjust maneuver sequences can be a complex process when the objective is to reach a specific target orbit, but because the objective of this SAPA is to simply change the orbit of the satellite to avoid a collision, the maneuver logic can be greatly simplified. Orbit Logic is researching and
developing simplified collision avoidance approaches (including attitude selection and burn duration computations) to reduce processing time and memory footprint requirements.

Because of the high priority nature of collision avoidance, any initiation of avoidance maneuver logic in this SAPA (due to a close approach buffer violation) will also cause this SAPA to trigger the APS interrupt mode (resetting the SAPA/MAPA planning cycle) to ensure that the avoidance maneuver can be executed as soon as possible.

J. Space Imaging SAPA Component

The objective of this SAPA is to perform space object imaging using an onboard optical sensor. The SAPA will monitor the message bus for space object imaging request events (that could originate from onboard sensors, another satellite or the ground) and plan imaging of the space object specified in the request. Imaging planning will be based on the latest spacecraft ephemeris, the space object estimated ephemeris or position, spacecraft attitude control characteristics, and onboard imaging sensor characteristics. This SAPA activity may rearrange routine imaging activities to meet a high priority space object imaging request need.

Unlike the SAR Ground Imaging SAPA which simply needs to select beam on/off times, this specialized planning agent will need algorithms to iterate with spacecraft or sensor attitude maneuver models in order to account for pointing of the sensor to acquire and track the space object during the configurable imaging duration. Orbit Logic will leverage its agile imaging spacecraft algorithms in the development of this SAPA.

K. APS Algorithms

Orbit Logic has over a decade of experience implementing and deploying planning and scheduling algorithms in the aerospace industry. During the SBIR Phase I Orbit Logic performed an algorithm trade study to specifically look at algorithms and their suitability for the limited processing resources available onboard for flight software. In addition, the study noted that different SAPAs (and the MAPA) may benefit from different algorithm types, depending on the kind of deconfliction or optimization required. During the Phase I algorithm trade study, it was determined that the best algorithm implementation for each SAPA is unique to the problem it solves. In Phase II, the researched algorithms were considered for each of the following prototype components:

- MAPA: The main purpose of the MAPA is to deconflict and coordinate the plans generated by the SAPA’s. The deconfliction will rely on a set of rules that are unique to the system. Therefore, rule based expert systems and search algorithms will be considered for implementation of the MAPA
- Collision Avoidance SAPA: A specialized orbit propagation and orbit adjust algorithm is required to perform collision avoidance. Expert systems, search based, and genetic algorithms will be considered for implementation.
- SAR Ground Imaging SAPA: Based on the results of the Phase I algorithm trade study, it was determined that a best-first-search algorithm will be used for the SAR SAPA to meet performance and memory footprint constraints.
- Space Imaging SAPA: The space imaging SAPA is similar in nature to the SAR Ground Imaging SAPA. Therefore various search algorithms including best-first-search, will be considered for implementation.

L. Mission Configurable Variables

The MAPA and SAPA components are configured through mission configurable variables (MCVs) that can be modified by table uplinks from the ground at any time. MCV modifications may be required to enable or disable the APS or individual SAPAs, or to change their operational parameters due to changing mission goals or a changing mission environment.

M. Telemetry and Command Interfaces

Emergent’s Ground Enterprise Management System (GEMS) ground monitoring package is used for sending commands to the APS flight software, ingesting the APS flight software telemetry to visualize results on custom display pages, and to orchestrate testing. Some sample GEMS displays are shown in Figure 11. GEMS has a drag-and-drop display builder to allow a user to quickly construct displays so that minimal time will be spent on ground monitoring development to allow more resources to be focused on the flight software development. GEMS displays give operators visibility into the system and provide the commanding interface necessary to control and configure the APS.
VI. Conclusion

Giving satellites the ability to make autonomous planning decisions will allow satellites to respond much more quickly to capture opportunities that might otherwise be missed. The APS MAPA/SAPA architecture for onboard planning supports flexibility to plan for different kinds of opportunities, keeps the system modular and efficient enough to be used in an onboard environment, and makes the system extensible to almost any kind of planning with the use of specialized autonomous planning agents (SAPAs).

APS, which assembles a planning system from multiple specialized components, is particularly important for onboard planning where limited processing resources compel the use of the most efficient algorithm types. Our design asserts that the most efficient algorithm type depends on the kind of planning problem being solved. With the SAPA component architecture, the most efficient algorithm can be applied to each planning domain without any inefficiency or conflict, and only components (and associated algorithms) for a particular program are included in the flight software for those satellites.

The APS modular and configurable planning system architecture could be applied to reduce the cost, schedule, and risk of implementing planning systems on various platforms, while at the same time making the resulting planning systems more agile to respond to dynamic mission goals and more efficient with the use of processing resources.

Acknowledgments

Portions of this work were performed under contract with the Air Force Research Laboratory (AFRL).

References