A Model-based Approach to Controlling the ST-5 Constellation Lights-Out Using the GMSEC Message Bus and Simulink

SERP 2005 Conference, Las Vegas, NV

Kenneth J. Witt
Senior Member Research Staff
Institute for Scientific Research
Fairmont, WV 26554

Robert Shendock
SGT Inc.
NASA Goddard Space Flight Center
Greenbelt, MD 20771

Jason Stanley
Member Research Staff
Institute for Scientific Research
Fairmont, WV 26554

Daniel Mandl
NASA Goddard Space Flight Center
Code 584
Greenbelt, MD 20771

Abstract – Space Technology 5 (ST-5) is a three-satellite constellation, technology validation mission under the New Millennium Program at NASA to be launched in March 2006. One of the key technologies to be validated is a lights-out, model-based operations approach to be used for one week to control the ST-5 constellation with no manual intervention. The ground architecture features the GSFC Mission Services Evolution Center (GMSEC) middleware, which allows easy plugging in of software components and a standardized messaging protocol over a software bus. A predictive modeling tool built on MatLab's Simulink software package makes use of the GMSEC standard messaging protocol to interface to the Advanced Mission Planning System (AMPS) Scenario Scheduler which controls all activities, resource allocation and real-time re-profiling of constellation resources when non-nominal events occur. The key features of this system, which we refer to as the ST-5 Simulink system, are as follows:

- *Original daily plan is checked to make sure that predicted resources needed are available by comparing the plan against the model.*
- *As the plan is run in real-time, the system re-profiles future activities in real-time if planned activities do not occur in the predicted timeframe or fashion.*
- *Alert messages are sent out on the GMSEC bus by the system if future predicted problems are detected. This will allow the Scenario Scheduler to correct the situation before the problem happens.*
- *The predictive model is evolved automatically over time via telemetry updates thus reducing the cost of implementing and maintaining the models by an order of magnitude from previous efforts at GSFC such as the model-based system built for MAP in the mid-1990's.*

This paper will describe the key features, lessons learned and implications for future missions once this system is successfully validated on-orbit in 2006.

1 Introduction

Limited budgets, increasing number of satellites flying and the continual increasing complexity of missions are driving the need for mission autonomy. This implies the need for autonomous management of shared resources between sets of heterogeneous satellites, especially during non-nominal situations. One approach to solving this problem is using predictive models that can automatically trigger actions to resolve problems before they occur. This approach is called Model Based Operations (MBO). MBO can save significant manpower and other resources throughout the lifecycle of a mission as the collection of satellites managed increases and become more complex.

MBO has heritage in previous efforts at the Goddard Space Flight Center (GSFC), beginning in the late 1980's. The Gamma Ray Observatory mission attempted to create a prelimary implemention that used time varying limits, which in effect, is a rudimentary MBO approach. Later the ALTAIR system was used to build models of the Microwave Anisotropy Probe (MAP) mission operational scenarios in the 1993 timeframe. ALTAIR used a top down comprehensive approach to model the behaviors of the entire spacecraft. This approach proved too costly to maintain due to the complexity of the model and the changing characteristics of the spacecraft subsystems. Thus we have evolved a more cost-effective, build-as-needed approach, that allows for a mission to grow its MBO component based on need, funding, and requirements.

2 Application

Space Technology 5 (ST-5) is a mission in NASA's New Millennium program, designed to validate new space technologies. In addition to the requirements defined to support routine mission operations, high-level mission operations requirements for ST-5 place emphasis on the following:

- Support for constellation operations
- Perform autonomous operations
- Minimize operations costs

Previous efforts at NASA have addressed the issues of cost minimization and autonomous operations in multiple ways with varying levels of success. Constellation operations are a new requirement for missions in the GSFC environment. After evaluating traditional application software systems previously utilized at GSFC, we found that enhancements were needed to meet the ST-5 requirements with available application software. In particular, significant changes in the software systems' structure and more software autonomy were needed in order to fulfill the ST-5 requirements.

A graphical representation of the ground system supporting ST-5 mission operations is provided in Figure 1. The system is composed of a collection of application software systems, each designed to meet a specific set of functional requirements. The final system design includes the features discussed in this paper, which have demonstrated reduced development and operational costs through the implementation of a common communication bus and a model-based operations approach.

Commands, directives and data are shared among the applications by use of an emerging technology developed at GSFC. The GSFC Mission Service Evolution Center (GMSEC) middleware bus provides the necessary means for communication among the applications for the design to support autonomous operations.

Model-based operations provides relief from the lower-level recurring analytical tasks traditionally performed by off-line mission planning and on-line engineering staff. This is done by predicting problems before they occur and sending alerts to enable the ground system to automatically take action to mitigate the problems before they occur . Unique to our approach, is the ability to be cost-effective; by only applying the predictive models to areas where doing so provided measurable returns. This approach contrasts most previous model-based operations approaches, which attempted to take a top-down universal predictive modeling approach. However, the top-down approach which quickly became overly complex. In the case of ST-5, the approach is applied to the active management of spacecraft constrained resources which for ST-5 include; on-board data storage /recovery, RF link quality and electrical power. This approach allows a mission to only model and use as much as the mission can afford with easy future extensibility if the need arises.

3 Real-time Object Modeling Executive

The infrastructure required to create an autonomous operation environment consists of several key pieces. First, a common communication structure must be in place. The systems utilizing the structure must be able to determine the proper activities and responses to events without user intervention. The system must also provide a mechanism to determine the given state, as well as predict the future state, of the mission being managed.

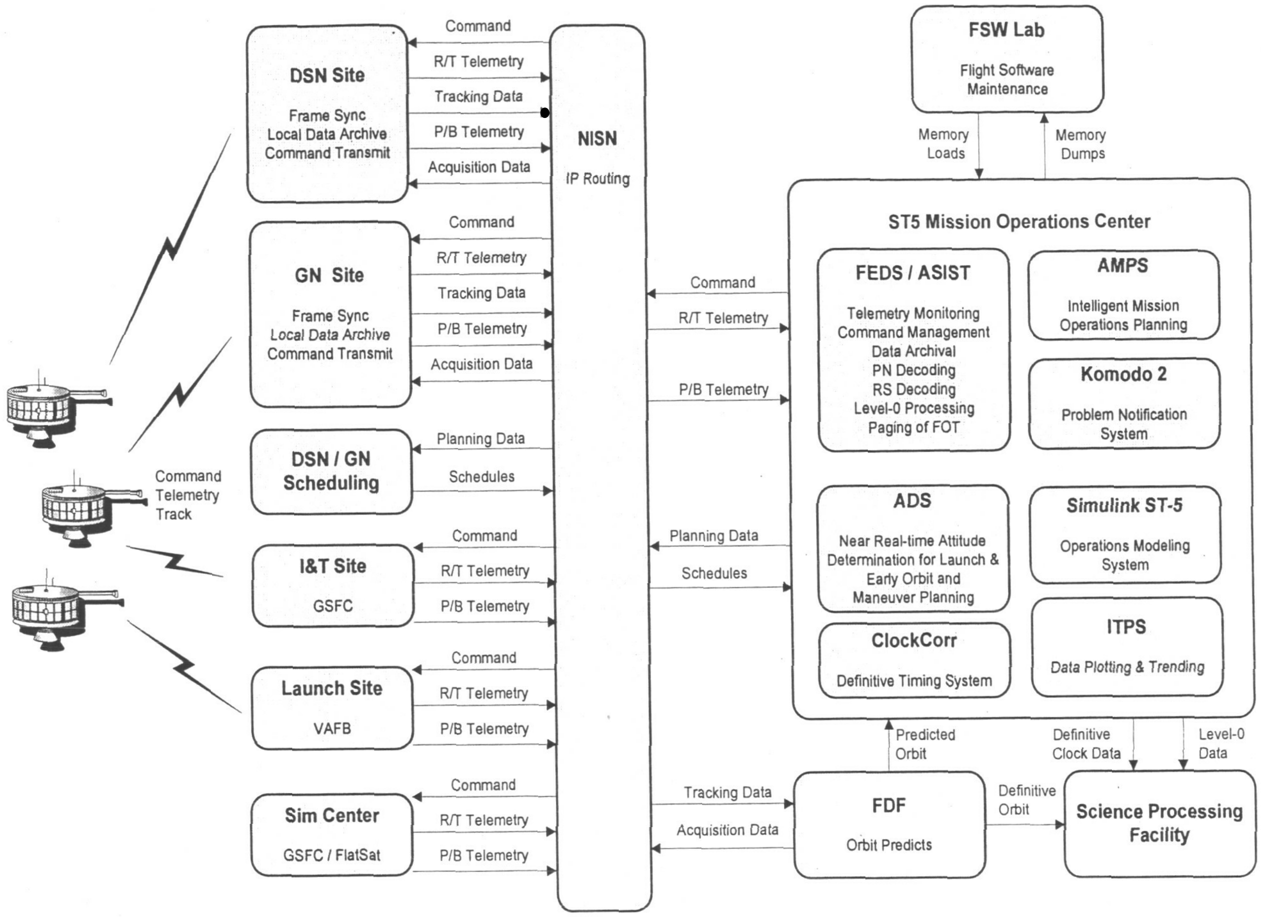

Figure1: ROME Web Interface, for ST-5

The Real-time Object Modeling Executive, or ROME, framework is designed to allow for operation centers to leverage the model-based operation concept. This concept allows for the use of spacecraft models in day-to-day mission support activities to predict spacecraft system and subsystem status. Models created during the engineering design phase, or those built to directly support a mission, can be easily integrated into an operational environment. The models can then be used to support autonomous or 'lights out' operations.

ROME utilizes XML based configuration files that allow operators and managers to rapidly configure the framework. As models are added or removed, the definitions, attributes, and output are described in an XML configuration file. The XML is used to describe the model's interaction with the bus. The XML describes both the telemetry mnemonics (spacecraft data) needed to drive the model, as well as how to package the results for publication back to the bus.

3.1 Communication Structure

ROME was built around the concept of a modular operations center. The various subsystems utilize a architecture that allows applications to publish and subscribe information with other applications on the bus. GMSEC is striving to eliminate the need for developing dedicated interfaces between operation center components. GMSEC has defined a standard set of messages that relate to operation center functionality. These messages can be customized to implement unique mission requirements and still maintain a standard communication protocol. See http://gmsec.gsfc.nasa.gov for more information specific to GMSEC.

GMSEC is built around commercially available middleware systems. These products range from a no cost Government- Off-The-Shelf (GOTS) provided option, through a fully redundant commercial system that allows for failover and guaranteed message delivery. Middleware can be interchanged with little to no impact on the GMSEC applications using the bus.

ROME utilizes the GMSEC bus to talk to its operational peers. The planning system, command and control system, notification system, logging system and others are built to, or retrofitted to be GMSEC compatible. Via the bus, directives are sent and received to manage the flow of data and communication between operation center systems.

3.2 System and Subsystem models

A mission's requirement to analyze subsystems will vary greatly in fidelity and granularity. Missions may need to manage or plan against a specific subsystem's parameters or may require comprehensive vehicle analysis and therefore use a detailed model of the entire spacecraft. The precision of a given model has little impact on its ability to be integrated into the ROME framework. A mission may also use a limited set of models to start and then grow into using more models as the mission matures or as needs become evident.

ROME is also designed to allow for its models to self-adjust, or tune themselves, over the lifetime of the missions. The performance characteristics of actual spacecraft hardware will change over time and with use. Solar panels, batteries, and link margins can degrade or improve during the course of the mission. A model that may work in pristine fashion on the outset of a mission may be worthless 3 weeks later. The ROME framework was designed with this in mind, so features are built-in to help adjust the characteristics of the models so that they accurately reflect the reality of the systems they are representing.

During contacts with the spacecraft, telemetry is collected via the bus and stored in a database. The telemetry is then packaged and sent to another Matlab-based model specializing in trending. The trending model is a method developed for each of the subsystem models that adjusts parameters which vary over time. The output of the trending is used to initialize the subsystem models as they execute in real-time. Thus the models are constantly being tuned to be as accurate as possible.

3.3 Modeling Environment

NASA has made a large investment in developing models for various aspects of mission development. Many of the models are built using Matlab and Simulink because of the flexibility and power afforded by the toolset. Models are also developed in more specialized tools such as Satellite Took Kit (STK), and Free Flyer. Models can, and are, developed throughout the mission lifecycle. Previously, these models were abandoned at launch. ROME strives to utilize these models in the operational environment, without incurring significant integration costs.

3.4 Interfaces

ROME provides two interfaces to view the results of its profile activities and to instigate offline plan validation. ROME has a dedicated console application, built on top of Microsoft's .NET framework as depicted in Figure 2. ROME also provides a web-based interface for user interaction as depicted by Figure 3. Both provide graphical feedback representing the results of the profiles for each spacecraft and each of its subsystems. The ROME framework leverages XML data structures for storing profile result data. The web interface graphing functionality leverages XML-based Scalable Vector Graphics (SVG) for portability across platforms and resolution independent presentation.

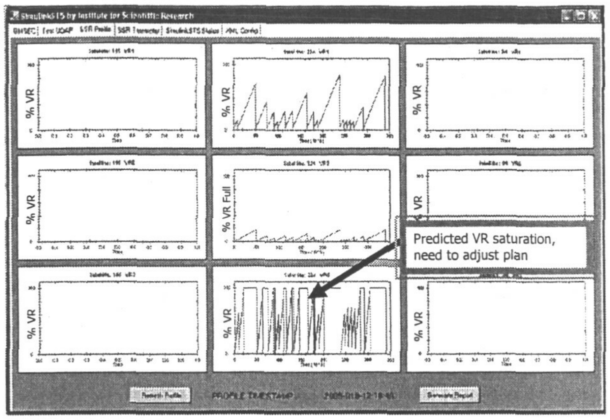

Figure 2: ROME Console Interface showing status of the ST-5 Virtual Recorders (VR)

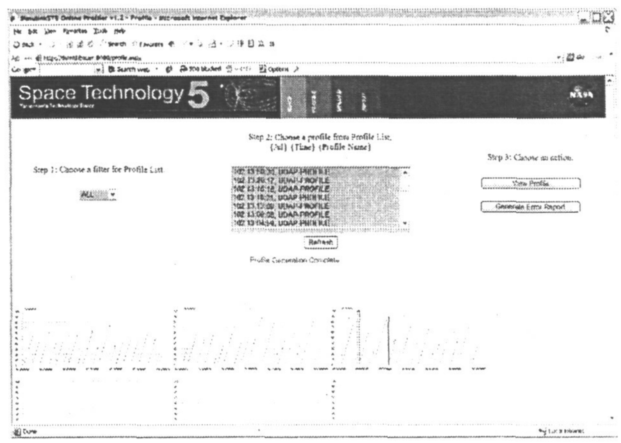

Figure3: ROME Web Interface, for ST-5

4 Operational Scenario

Using the ST-5 mission for demonstration, we will illustrate the typical operational scenario. The mission is a three-satellite constellation designed to study the magnetosphere. (See http://st5.gsfc.nasa.gov)

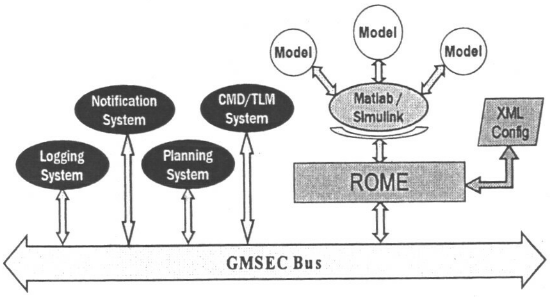

Figure 4: ROME Framework using GMSEC

The ground operation system, as depicted in Figure 4, consists of ROME; a planning system, Advanced Mission Planning and Scheduling System (AMPS); a telemetry and command system, Advanced Spacecraft and SystemsTest System (ASIST); an alert/ notification system, Komodo II; and a logging system, GMSEC Reusable Event Analysis Toolkit (GREAT). All of these systems communicate via the GMSEC bus.

The ST-5 mission has chosen to implement models for its Solid State Recorder (SSR), communications subsystem, and its power subsystem. All of the models are implemented using Matlab/Simulink. Each of the subsystems is modeled for each of the spacecraft, effectively yielding 9 models for the mission.

In this operational environment, ROME functions in two modes. The first is to perform a validation and verification of a candidate activity plan. The second is to provide real-time feedback as to system state and health. Figure 5 depicts the functional flow of the ROME framework.

4.1 Validation and Verification of Activity Plans

To perform its validation role, when an ST-5 ground system operator generates a plan, ROME will verify the plan by propagating the models from an a-priori state for the plan's duration. Resultant information is fed back to the operator. Significant errors may result in the regeneration of the activity plan. This cycle will continue until all significant risk of resource contention has been removed. At this point the plan is published,

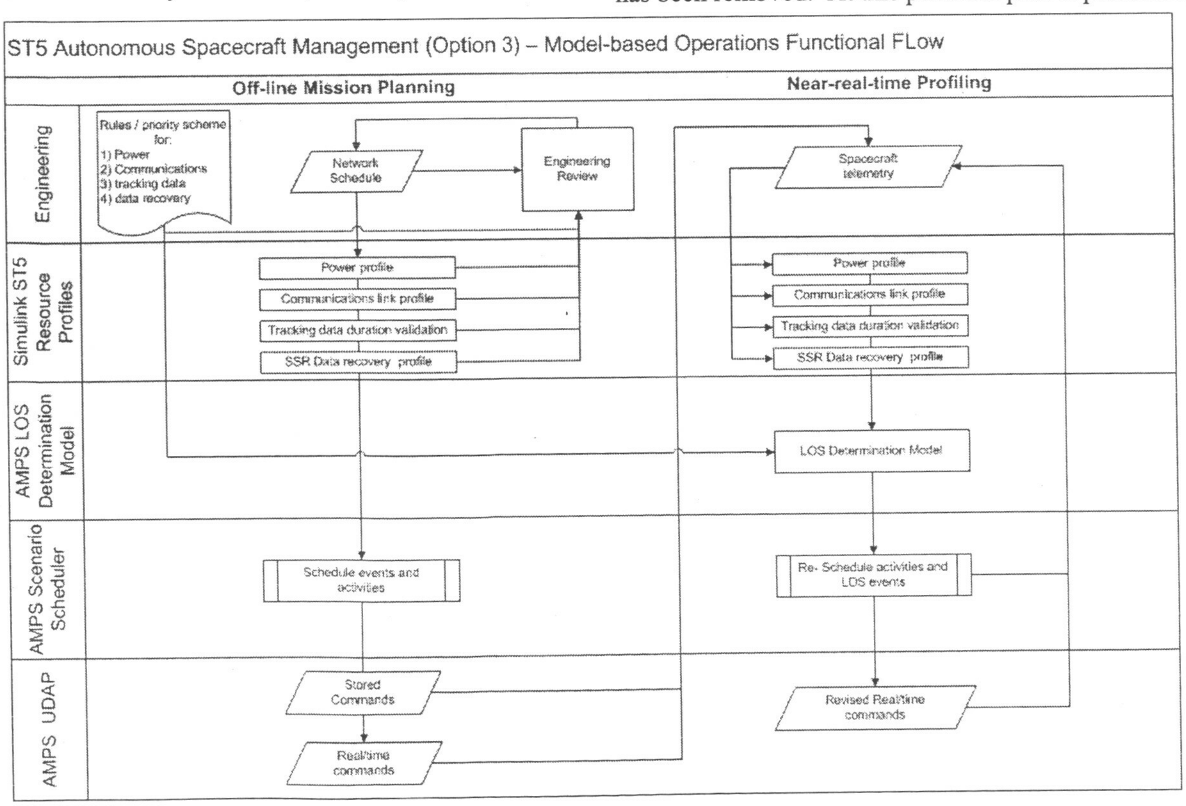

Figure 5: ROME Operational Modes

via the bus, to all interested consumers.

4.2 Real-time Situational Awareness

After AMPS produces and validates the plan, it is published to the bus for consumption by subscribing systems. ROME utilizes this plan to perform prediction of spacecraft subsystem states. The activity plan constitutes all actions and timelines within the mission framework. The satellites perform actions based upon a plan generated by AMPS.

Based on the activity plan, when a satellite is scheduled to start a contact, AMPS will publish a directive to the bus to inform all applications of this event. ROME will then generate a request directive to receive spacecraft telemetry data. ASIST will then continuously publish the requested telemetry values. As the contact progresses, the model's states are updated utilizing the telemetry stream.

At a predetermined point during the contact, AMPS will issue a directive requesting ROME to profile one or more subsystems. ROME will propagate the subsystem state over the duration requested. The results are packaged and published to the bus.

AMPS will consume the results and make plan alterations to compensate for any predicted resource utilization overflows.

As the contact completes, AMPS will notify all subsystems of the contact's conclusion via a GMSEC directive. Upon this notification, ROME will request the cessation of the GMSEC telemetry stream.

5 Lessons Learned

The key lessons learned thus far are as follows:

(1) A bottoms-up approach can be more cost-effective than a top down approach for modeling operational activities. This is because the bottoms-up approach, as taken on ST-5, focuses only on key areas that can provide a positive return on the time invested in building the models.

(2) A generic message bus such as the GMSEC bus is essential in making this approach easily scalable.

(3) Self-maintaining models are essential to avoid the need for a large engineering contingent to maintain the models.

(4) Making use of common modeling tools such as MatLab/Simuink together with the bottoms-up approach (taken on ST-5) makes each of the models more independent and reusable for future missions.

6 Future Implications

6.1 Modeling lowers costs

Future implications can be categorized into short-term and long-term implications. In the short-term, because of the predictive and autonomous nature of MBO, many of the crises that can occur during operations are averted in a preemptive manner. Problems are predicted days in advance giving operators time to make sound resolutions. This means that operations staff deal with major problems during normal work hours, and the Mission Operations Center (MOC) no longer needs to be manned 24x7, thus lowering operational costs.

Furthermore, the level of automation and autonomy is raised to a higher level. Whereas some mission use lights-out automation to automate traditional console procedures to manage their satellite, the activities performed by the Flight Operations Team (FOT) engineers is still required. There engineers respond to non-nominal situation. On ST-5, we will use MBO to experiment with automating the activities that the FOT engineers traditionally perform.

In the long term, one can envision smart constellation components, each having a predictive modeling component that anticipates problems and autonomously optimizes its future activities. Furthermore, each component performs peer-to-peer negotiations with other components to fully optimize the usage of shared resources throughout the entire system. An example might be that a data recorder on one satellite is almost full due to an unanticipated science event thus requiring an image acquisition. The data recorder might negotiate over a message bus with the ground station to raise its own priority over another prescheduled satellite to avoid losing key data.

6.2 Multi-level Modeling

Models can be derived from many potential sources depending on mission need and requirement. The ALTAIR system mentioned earlier required a large number of very knowledgeable engineers to maintain its behemoth models. ROME allows for missions to adopt models developed by even students who may understand a subsystem without needing a comprehsive grasp of the entire platform or spacecraft. Models created by design engineers will have an extended life in the MOC due to the increased ease in abstracting their box model into an operational environment. The scalable, flexible nature of MBO and ROME offer great flexibility to operations managers.

Reuse of the models will have a forward and backward benefit to NASA missions in the general sense. It is anticipated that the reuse of the ROME models will be inexpensive for operations. Operations can feedback to engineers the accuracy of the models for future. Once MBO is more widely adopted, the closed loop feedback between operations and engineering will be invaluable to NASA in moving the state-of-the-art of model-based operations forward.

6.3 Message bus extensibility

Further enhancement of of model-based operations will be enabled extending the message bus to seamlessly include the spacecraft. Thus some of the modeling activities will occur onboard the satellite. Some of these experiments are being conducted through projects like Livingstone (See http://ic.arc.nasa.gov/partner/files/Livingstone_2.pdf). This approach will make the mission system more responsive during situations requiring rapid response. The implication is that problems and anomalies can be addressed and corrected much faster than having to wait for ground contact and ground controlled resolutions. In this mode, ROME could delegate some of the modeling to an onboard system and then integrate higher-level alerts and analyses.

6.4 Rudimentary Machine Learning Capability to Offset Mission Complexity

As NASA missions become increasingly complex, MBO and ROME will help to abstract operational activities and take care of many low-level details automatically. Since ROME is designed to allow for models to self-adjust over time, much like a commuter learns the best routes to a destination based on experience over time, so might satellite systems learn how best to achieve mission goals over time.