

Mission Planning for a Formation-Flying Satellite Cluster

John L. Mohammed

Stottler Henke Associates, Inc.
1660 South Amphlett Boulevard, Suite 350
San Mateo, California 94402
Mohammed@shai.com

Abstract

The formation-flying satellite cluster is a new paradigm for space-based surveillance and remote sensing. In this paradigm, several small satellites fly in close formation and coordinate their activities so that through sparse-array interferometric and synthetic aperture techniques the cluster can effectively operate as though it were a much larger monolithic satellite. Management of such clusters will require increased automation for planning and scheduling, both in pre-flight mission planning and during operations. This paper describes work aimed at defining and meeting the special planning and scheduling requirements of the formation-flying cluster.

Introduction

A satellite cluster is a group of satellites that fly within very close range of each other (e.g., 250m-5km). These satellites coordinate their activities, so that they can use sparse array interferometry and synthetic aperture techniques to simulate a single, very large satellite. The cluster operates as a “virtual” satellite with a very large effective aperture, without the need for the heavy infrastructure that would be required to have a monolithic satellite with the equivalent aperture. The cluster approach has many advantages over a single large satellite:

- ✓ Each spacecraft is smaller, lighter, simpler, and simpler to manufacture;
- ✓ Economies of scale enable a cluster of many satellites to be less expensive to manufacture than a single satellite;
- ✓ The cluster can adapt to the failure of any individual satellites, and failed satellites can be incrementally replaced;
- ✓ The cluster can reconfigure the orbits of the satellites in the cluster to optimize for different missions.

A constellation of clusters would enable whole-earth coverage from low earth orbit and/or continuous coverage

for specific theatres. However, clusters and constellations present a significant challenge to current methods for the management of space-based assets. Current practices are too labor-intensive and would not scale well to the large numbers of satellites that would have to be managed. Automation will have to play a much larger role in planning and operations, and tools for automation will have to be knowledgeable about the unique characteristics of the formation-flying satellite cluster. The number of satellites to be managed, and the wider range of parameters that can be optimized demand new tools for planning, scheduling and optimization.

This paper describes a project concerned with the development of a mission planning system for clusters of formation-flying satellites: **Spacecraft Cluster Automatic Planner/Scheduler (SpaceCAPS)**. The project’s focus is on planning and scheduling for payload management. The paper describes some of the planning and scheduling requirements for cluster payload management, and describes a system for optimization of the mission plan for a space-based radar surveillance system. The discussion is grounded by reference to the Air Force TechSat 21 program.

The TechSat 21 Program

The TechSat 21 program (Technology Satellite of the 21st Century) is a coordinated effort of several Air Force Research Laboratories directorates to study a variety of application missions for the satellite cluster concept. The initial focus of the program is on Ground Moving Target Indication (GMTI) and Synthetic Aperture Radar (SAR) imaging.

Each micro-satellite in the cluster transmits radar pulses that are orthogonal to those transmitted by every other satellite in the cluster, and each detects and coherently combines the returns from every satellite in the cluster. In this way, the micro-satellites in the cluster form a large but sparse coherent array, enabling collection of angle- and time-of-return data with an effective aperture equivalent to the separation between the micro-satellites.

The vision for the deployed system is a constellation of “virtual” satellites, with each virtual satellite being a cluster

of micro-satellites. The numbers are still to be determined, but according to one published account (Martin and Stallard 1999) there would be 35 virtual satellites and 5 spare virtual satellites. Each cluster would contain eight micro-satellites flying within 250 meters of each other. The constellation size is motivated by a desire for full-earth coverage and/or continuous coverage in two theatres. The number and spacing of the micro-satellites in each cluster is motivated by performance requirements for the GMTI mission. Other proposals mention cluster sizes up to 16 micro-satellites, and other missions (such as passive geolocation) require larger separations (up to 5km).

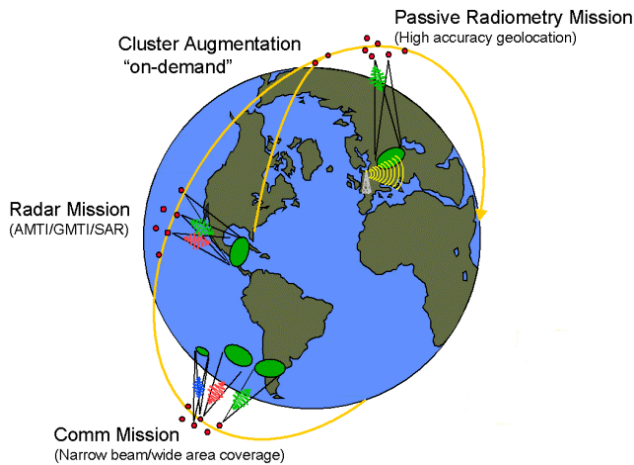


Figure 1 TechSat 21 Mission Concept (image from TechSat 21 Program Overview (AFRL 1998)).

A one-year technology demonstration flight is planned for launch in November 2003. This flight will consist of one cluster of only three micro-satellites. This single cluster will reconfigure the orbits of its micro-satellites for different phases of its mission, which will test different mission applications.

Distributed Space-Borne Planning

The SpaceCAPS project is ultimately concerned with a broad spectrum of planning and scheduling issues pertaining to a constellation of formation-flying satellite clusters. These will include medium- and long-term advance planning and day-to-day dynamic scheduling, using a combination of ground-based and space-based software planning and scheduling agents. However, the initial focus of the project is optimization of the mission plan using a ground-based mission planner. This mission planner would be used to plan and schedule the entire one-year mission for the first technology-demonstration flight of the TechSat 21 program. The objective is a system that

can incorporate detailed, cluster-specific constraints into the planning and scheduling of payload activities for the cluster.

The following sections briefly describe the architecture of the overall system and the issues it must resolve, then they describe the mission-planner system in greater detail.

The focus of the onboard software components is to enable autonomous operation. The onboard planning and scheduling system is an agent-based system that interacts with other onboard agents encapsulating the software for flight dynamics, guidance and navigation, power management, etc. Our architecture employs a dynamic hierarchical social organization to combine the efficiencies of hierarchical task delegation with the robustness afforded by dynamic reorganization. The architecture assumes the existence of satellite cross-link communications between clusters. This is described further in (Richards, Houlette and Mohammed, 2001).

The system plans at three levels of abstraction: constellation, cluster, and individual micro-satellite. Planning at the constellation level focuses on task delegation: selection of the best cluster for each mission task, based on windows of opportunity, performance predictions, costs and available resources. Depending on the degree and reliability of inter-cluster communications available, this could employ centralized planning or distributed contract-net negotiation techniques. Costs would include the delta-V penalty needed to re-phase a cluster for a specific task.

Planning at the cluster level focuses on coordination of the micro-satellites. Planning and scheduling at this level can rely on the availability of good inter-satellite communications, because these are required for relative orbit maintenance (formation-keeping), and the time-synchronization and relative-distance measurements needed to support coherent combination of SAR data. On-board processing of observation data would also require high-bandwidth inter-satellite communications for data sharing to enable coherent data combination.

At the individual micro-satellite level, the focus is on planning and scheduling of activities that do not require coordination, and supporting cluster-level planning by distilling information regarding local onboard resources. This level of the planning system is also responsible for the synthesis of detailed commands constituting directly executable procedures. This level of the planning system interacts with the spacecraft executive so that planning and execution can be interleaved (Chien, 1999).

The TechSat 21 technology demonstration micro-satellites will host agent-based software enabling the cluster to be managed as a single virtual satellite. However, the capabilities for planning and scheduling are likely to be very modest. Thus, at least initially most planning and scheduling will be performed by software on the ground. Nonetheless, wherever the planning and scheduling system is

hosted, it will have a distributed design that anticipates possible onboard deployment. This will ensure a migration path to space-borne platforms, and ensure that ground-based software has components that mirror the space-borne components of the system. This approach is reflected in our design for the ground-based mission planner, which is designed to optimize the payload schedule for an entire mission.

Payload Schedule Optimization

The initial focus of the ground-based mission planner is optimization of the quantity and quality of observations that can be made during the entire mission life, taking into account detailed constraints regarding resource usage, viewing geometry, downlink opportunities and bandwidth, and interactions with bus activities. The planner creates a detailed schedule that attempts to maximize the number of target observations and the quality of the observations while distributing the observations as evenly as possible among all the identified targets according to preference.

Target Selection

Scheduling for any satellite whose mission involves observations of and/or communication with the ground must take into account “accesses”—windows of opportunity defined by when the satellite is in view of the target on the ground. These are determined by the precise orbit that the satellite is in. For satellite clusters, one must also take the relative positions of the individual satellites into account when assessing viewing geometry.

Satellites cannot fly side-by-side in close formation without expending a great deal of fuel. “Formation-flying” satellites actually fly in closely related stable orbits. Unless the satellites are flying directly behind one another, the only stable orbits involve relative elliptical motion. Thus, the geometric configuration of the satellites is in constant periodic flux. When an observation requires coordinated action by all satellites in the cluster, not only must all the satellites be in view of the target at the same time, but also their relative positions should satisfy the requirements for good observational geometry.

If all observations employ all the satellites in the cluster, then one can try to optimize the orbits to maximize the viewing geometry for the greatest number of targets most of the time. (Kong et al., 1999) reports on a study that employed the Cornwell metric in an attempt to select orbits that maximize the total quality of all observations. This study measured the predicted quality of observations for several targets at several points in the cluster’s orbit.

The mission planner must also use a performance predictor to decide which targets should be observed and when. There will often be several targets in a position to be

observed at close to the same time. However, resource constraints will limit the number of targets that can be observed. For the demonstration flight especially, the objective will be to maximize both the number and quality of the observations made. This implies that the planner must select the targets to observe based on the predicted performance.

Scheduling in this domain also differs from most other domains in that there is not a finite, pre-determined set of tasks to be scheduled. In most other domains, tasks are either enumerated, or they are periodic (e.g., schedule at least three instances of this task each day). In this domain the objective is to schedule as many observations as possible within a given timeframe, subject to all the constraints. Further, there may be many targets, so another objective is to distribute the scheduled observations as evenly as possible among all the targets (subject to declared preferences among targets).

Sensor Allocation

Some observations can be performed with a subset of the satellites in a cluster. In such cases the mission planner must select which satellites to employ for such observations. This decision must also consider the effect of viewing geometry on performance as well as the availability of onboard resources (such as memory and power).

Downlinks

Synthetic aperture radar techniques generate very large amounts of data (the same is true for hyperspectral instruments). For example, the TechSat 21 micro-satellite (in a cluster with two other satellites) collects 9.6 Gigabytes (GB) per satellite for a two-minute observation. With overhead, this means that the total amount of data that must be downlinked from all three satellites is 33.5 GB. At the expected downlink bandwidth of 150 Mbps, this will take approximately 32 minutes. In low earth orbit, the micro-satellites are in view of a ground station for approximately four minutes at a time, about four times a day. This means it can take two days to downlink the data from one two-minute observation (AFRL, 2000).

Thus, onboard memory capacity, downlink accesses and downlink bandwidth are the constraints that most limit the number of experiments that can be performed. The mission planner must select the ground station for each downlink taking care to observe constraints regarding contention for ground station resources.

SpaceCAPS Mission Planner

Figure 2 displays a very high-level block diagram of the architecture of the ground-based mission planner. The planner obtains ephemerides and time windows for accesses

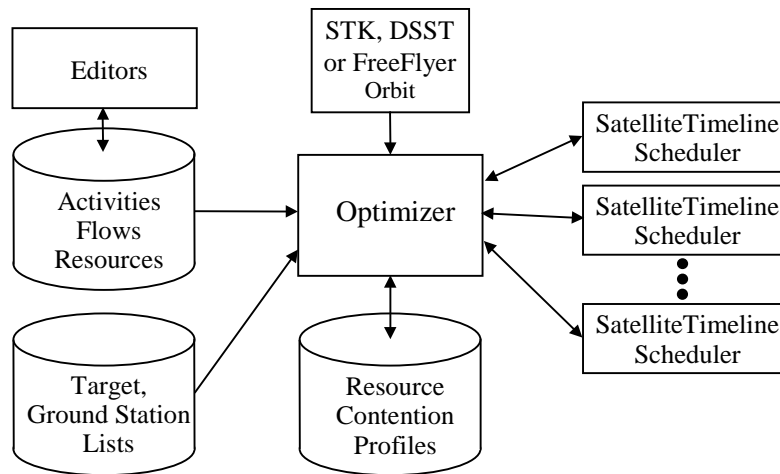


Figure 2 High-level architecture of SpaceCAPS Mission Planner.

to targets and ground stations from a third-party commercial-off-the-shelf (COTS) orbit propagation system such as Satellite ToolKit (STK) from Analytical Graphics, Inc., FreeFlyer from AI Solutions, Inc., or DSST from Draper Labs. The Phase I SpaceCAPS prototype interoperates with STK via TCP/IP network streams using STK/Connect.

The planner employs a separate “timeline” scheduler to perform detailed scheduling of the activities onboard each satellite. These schedulers correspond to the planning and scheduling agents that might be migrated onto the spaceborne platform. They perform detailed scheduling of the flows needed for payload experiments using heuristic constructive scheduling techniques. The heuristics concern the order in which different activities should be scheduled, and the scheduling technique that should be used for each. A centralized schedule optimizer component calls upon the individual satellite timeline schedulers in an iterative repair algorithm designed to optimize the number, quality, and distribution among targets of the payload experiments.

Activity Model

The system’s knowledge of what activities can be performed, what resources they require and what conditions and constraints must be satisfied is stored in a knowledge repository called the *Activity Model*. Activity descriptions can be hierarchical. An activity can be decomposed into a set of sub-activities with temporal constraints among them. This can include sub-activities to be executed in parallel on separate micro-satellites within a cluster. A composite activity is called a *flow*. The description of an activity may include heuristic information regarding how the constructive scheduler should schedule the activity (i.e., which of several scheduling methods should be used).

The activity model describes *types* of activities and resources. A request to schedule an activity specifies the activity type, and results in one or more *instances* of the

activity being inserted into the schedule.

Resource descriptions can also be composite, defining a set of related resources, such as all the resources that are local to a particular micro-satellite. Composite, or structured, resources are required to enable the modeler to specify that when selecting separate resources for a specific set of related activities, the planner should select related resources. For example, if a specific sensor (micro-satellite) is selected for an experiment, then the payload memory onboard that same micro-satellite should be selected to store the experiment data.

In specifying the resources needed by an activity, the system enables the modeler to specify a restricted variable rather than a specific resource instance. The restricted variable is bound during the planning/scheduling process to an instance of the specific resource type to which the variable is restricted. By this method, the resource allocation decision is delegated to the planner. This is how, for example, the sensor allocation decision is left to the planner to decide based on performance prediction and resource availability.

If the resource type is a composite resource, then the variable is also composite, and the modeler can make requests on the sub-resources of whatever instance will be bound to the variable. For example, the activity may require a resource that is not only a satellite, but also a satellite that has at least 10 megabytes of payload memory available. The modeler names variables so that they can be referred to by name in subsequent requests within the same activity or flow.

Editing of the activity model is supported by an object-oriented editing environment that treats the types and instances of activities, flows, resources and variables as objects with attributes and composite structure. The editors employ forms, trees and direct-manipulation graphics to enforce syntactic and certain semantic constraints, freeing

the modeler to focus on the content rather than the form of the modeling knowledge.

Optimization Algorithm

The mission planner uses a combination of constructive and iterative repair techniques. It uses constructive techniques to build an initial schedule that satisfies all constraints except those stemming from contention for resources among observations at the same priority level (resource contention constraints between observations and bus activities or higher-priority observations are honored). This schedules an observation for every target at every opportunity. The observations are not actually scheduled. Instead, the system determines the profiles of resource usage required by each potentially schedulable observation and these are stored in a *resource contention profile* database. The database detects conflicts stemming from contention for resources among the desired observations.

The iterative repair technique then iteratively reduces the number of resource contention conflicts by selecting an observation to be adjusted or removed, until there are no more conflicts.

During both phases, the system can call upon a performance predictor such as the Cornwell metric to obtain a quantitative measure of the quality of each observation. This is used to choose the optimal time within each access window to perform the observation, and the best subset of sensors to use. During iterative repair, the selected observation may be shifted to a sub-optimal time within the same access window or a sub-optimal set of sensors if that will reduce resource contention. If adjusting the observation cannot reduce resource contention, the observation is removed.

On each iteration of the iterative repair phase, the planner selects the observation to adjust using a heuristic selection criterion that incorporates four criteria: the number of conflicts (select the observation with the most), the quality of the observation (select the lowest), the rarity of the observation (select the observation whose target has the most scheduled observations relative to the total desired for that target, if any), and the preference (select the lowest).

The algorithm distinguishes between “preference” and “priority.” Priority is treated as an absolute ordering criterion. Targets with lower priority will be inserted into the schedule only after all targets with higher priority have been scheduled. On the other hand, preference is treated as one dimension in a trade-off space that includes quality and rarity: the planner can choose to include several lower-preference, higher-quality or rarer observations over a higher-preference, lower-quality observation.

The space of possible schedules is far too large to search exhaustively (planning and scheduling are NP-complete problems). However, by starting with the best possible

version of every possible experiment, then heuristically adjusting or removing the least desirable ones to eliminate contention for resources, the algorithm determines a conflict-free plan and schedule that heuristically optimizes the number, quality and distribution of experiments.

Conclusions

Formation-flying clusters of micro-satellites have several technical and economic advantages over monolithic satellites, but they also pose new technical challenges. Among these is the need for increased automation in mission planning and operations, and the need for planning and scheduling systems to be aware of new constraints and optimization criteria particular to cluster management. The SpaceCAPS project is identifying these new constraints and developing techniques to address them.

Acknowledgements

The author gratefully acknowledges the assistance provided by Lance Self, Paul Zetocha, Steve Fiedler, David Martin, and Maurice Martin. The work reported in this paper was supported by SBIR Phase I contract number F29601-00-C-0149 awarded by AFRL.

References

- AFRL, 1998. TechSat 21 Program Overview. Air Force Research Laboratories Space Vehicles Directorate. <http://www.vs.afrl.af.mil/VSD/TechSat21/>.
- AFRL, 2000. TechSat 21 Command, Control and Communications Planning Document, V1.01 (Draft), May 2000.
- Chien, S.; Knight, R; Stechert, A.; Sherwood, R. and Rabideau, G. 1999. Integrated Planning and Execution for Autonomous Spacecraft. In Proceedings of the IEEE Aerospace Conference (IAC), Aspen, CO, March 1999.
- Kong, E.M.; Tollefson, M.V.; Skinner, J.M.; and Rosenstock, J.C. 1999. TechSat 21 Cluster Design Using AI Approaches and the Cornwell Metric. In Proceedings of the 1999 AIAA Space Technology Conference and Exposition, Albuquerque, NM, 28–30 September 1999. AIAA-99-4635.
- Martin, M. and Stallard, M.J., 1999. Distributed Satellite Missions and Technologies — The TechSat 21 Program. In Proceedings of the 1999 AIAA Space Technology Conference and Exposition, Albuquerque, NM, 28–30 September 1999. AIAA-99-4479.
- Richards, R.A.; Houlette, R.T. and Mohammed, J.L., 2001. Distributed Satellite Planning and Scheduling. Submitted to 14th Florida Artificial Intelligence Research Symposium, FLAIRS-2001, Key West, FL, 21–23 May 2001.