Operational Multi-Mission Spacecraft Operations On-line Scheduling and Automated Execution

Pierrick GRANDJEAN, Antoine PITIE

EADS ASTRIUM
31, avenue des Cosmonautes
31402 Toulouse, France
Tel: +33 5 62 19 67 81
Fax: +33 5 62 19 69 64
email: pierrick.grandjean@astrium.eads.net

Abstract.

Improving operations efficiency and sharing ground resources among a large number of spacecraft, such as for satellite constellations or multi-mission facilities, has become a major goal of satellite operators and Space Agencies.

Ways to operations efficiency and autonomy are through automation, standardization and sharing of ground resource between several spacecraft or missions. However, spacecraft commanding still requires high levels of safety, reliability and flexibility. These needs are often felt to be exclusive since they require on-line scheduling. We have developed an approach that makes on-line scheduling an operational reality, by integrating resource allocation and task planning (i.e. Scheduling) with schedule execution and supervision within a consistent framework, in short-loop interaction.

This integration allows the operations schedule to be quickly and safely updated when new requests are introduced or when unexpected events happen, so as to always make maintain a consistent and safe use of available resources (antennas, equipment, network, ...). Short-loop re-scheduling also represents significant reduction of workload as compared to manual planning.

This approach has been implemented by EADS Astrium on several missions, some being already in operations, such as the INTELSAT Control Center that runs highly automated operations for up to 30 satellites, and the HELIOS II observation satellites user segment. The resulting operational software, named TIMELINE, has now significant operational and functional maturity.

Potential applications include Galileo, the future European constellation navigation system, and multi-mission ground stations and operations schedule management systems for Space Agencies.

1 Introduction

Improving operations efficiency has become a major goal of satellite operators and Space Agencies, while maintaining a high level of safety, reliability and flexibility.

This is felt to be achievable through evolutions of current concepts of operations:
Operations automation: relieving operator from low-level, routine, operations.
Operations standardization: operating heterogeneous satellites or missions within a common operation concept.
Sharing of ground resources: Sharing resources, such as antennas or equipments for operating several spacecraft or for backup purposes, enables to optimize resources use and to reduce equipment idle time.

Coordinating remote operations on a common set of resources then becomes a common need for Multi-Mission science or observation control centres, as well as for Constellations and satellite fleets operations. Such coordination requires specific software solutions, because the delay between each individual operation is reduced, especially when the number of spacecraft becomes significant. In addition, resource sharing means new constraints between operations, which have to be taken into account from the planning stage up to the real-time execution processes.
In addition, the very high level of safety and service availability for Space Systems require operations flexibility and reactivity that can be satisfied by dynamic schedule management and automated schedule execution.
In the past, automated schedule and resource management was traditionally feared in space systems, because it was associated with long computation times, and felt to slow down operation processes or to forbid direct and emergency commanding. In the same way, automation was often felt to reduce operation flexibility and sometimes to reduce operations control.

We have developed an approach where **scheduling optimization and execution automation are fully integrated** in the same application. This allows performing **optimization of resource management**, to relieve staff from routine tasks and painful planning, while keeping **a high reactivity and reliability** to ensure safe operations and keep **operation flexibility** to handle critical situations.

Resource management then becomes compatible with real-time operations and also improves operations efficiency.
This paper focuses on the interactions between planning and execution, starting by analyzing operational requirements for dynamic scheduling and schedule execution in multi-satellite or multi-mission ground segments.

2 Dynamic scheduling requirements

2.1 Distributed Schedule Management is multi-mission ground systems

Automated schedule management is distributed in most large ground systems, and even more when sharing facilities between several missions. Different levels are:
· central schedule planning stage, interacting with service users (i.e. spacecraft control and mission centers) and resource managers (i.e. ground stations)
· local schedule management execution and monitoring, by service users and resource managers

2.2 Operations scheduling requirements

In multi-satellite control centers, tasks are mutually constrained by standard scheduling constraints and by more specific constraints. Automated scheduling takes the whole set of constraints into account in order to assign resources to tasks and compute their best start time whenever it is necessary, and solve conflict over resources according to a priority scheme.
Standard constraints are for instance:
· Unary resource sharing between tasks (e.g. antennas, staff, or on-board equipment that can be used by only one task at a time)
· Absolute and relative time constraints between tasks and events (e.g. earliest start date, latest end date and precedence constraints)
· External time constraints (e.g. visibility periods for Low or Medium Earth Orbit satellites, or for geo-stationary satellites in Launch and Transfert orbit phases, or eclipse periods)
In addition to these standard constraints, satellite operations call for more specific constraints that are more difficult to handle. For instance, transition delays must be allowed between tasks for the set-up, configuration and release of ground equipment, including antenna pointing and lock on spacecraft. Also incompatibility between resources is often a constraint (i.e. network equipments that cannot run at the same time).

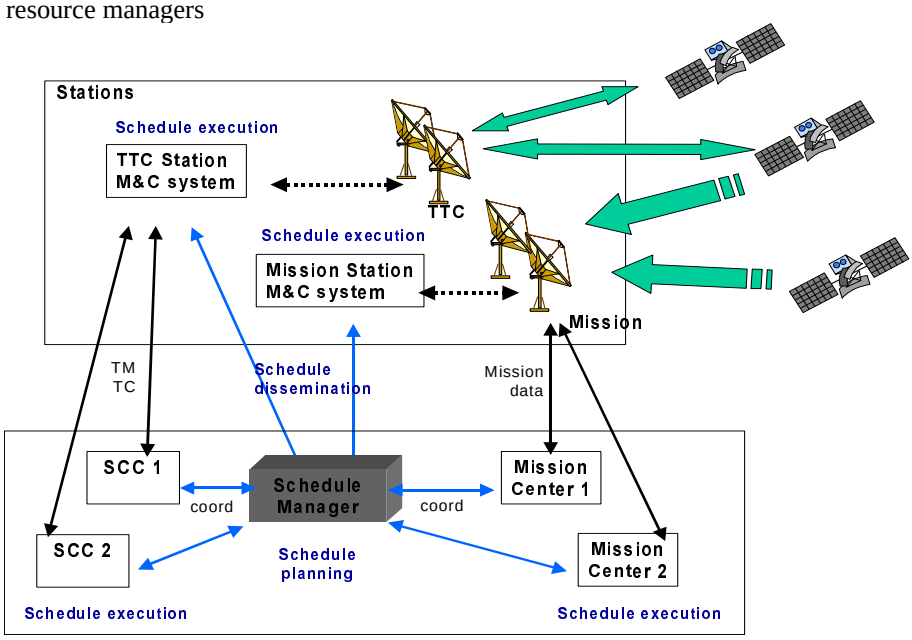

Figure 1: Multi-mission Schedule Management system architecture

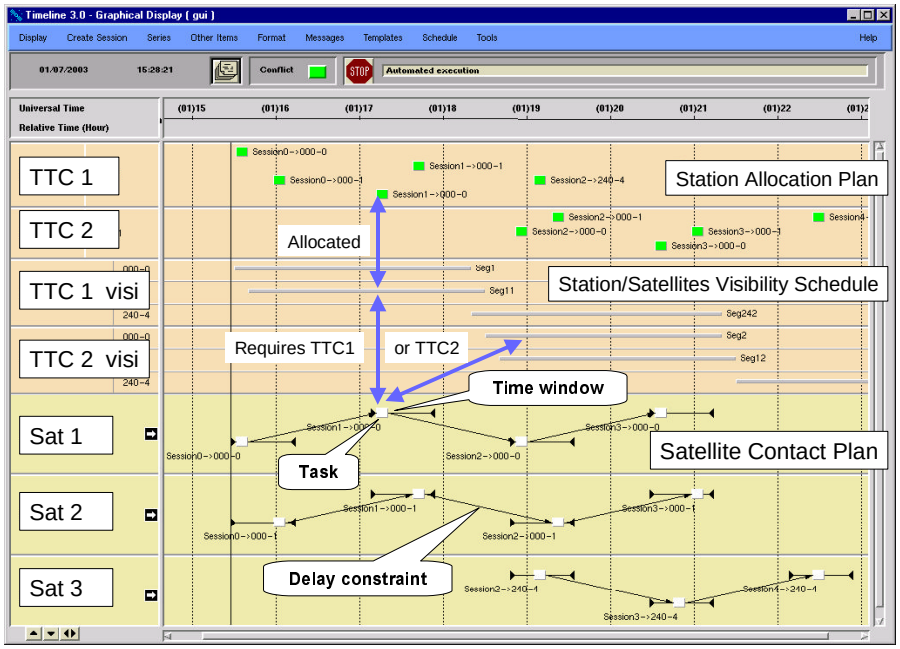

Figure 2: Example of multi-mission conflict-free schedule.

The bottom part shows the tasks and their temporal constraints. Each horizontal slice corresponds to one mission or satellite. The middle part shows the visibility schedule of each satellite vs. each station.
The above part shows the current assignment of Ground Resources to each task.

A particularly important aspect of operational scheduling is the capacity to always produce an executable and consistent schedule even when the whole set of constraints cannot be satisfied, rather than simply issuing a failure status as is usually the case for off-line planning tools. This must rely on operational rules such as priority ranking between tasks and constraints (constraint relaxation).
In the partially satisfied schedule, lower priority requests that remain unscheduled must be clearly identified to allow operators to resolve these conflicts manually by dialoguing with Users for modifying their requests (e.g. by extending or postponing the execution temporal window of tasks).

2.3 Execution and control requirements

Tasks must be started at their scheduled start time and then their progress must be monitored until their termination. This must support automated tasks executed by applications and also manual tasks whose progress and termination are indicated by operators from a User Interface.
An important feature for operators in Control Centers is the capacity to manually control task execution (task acknowledgment, task abortion, restart, etc.) and to edit task properties (duration, priority, etc.).
Also, some advanced monitoring systems can automatically trigger the scheduling and execution of new tasks to respond to detected situation. When the situation is critical, time-to-execution requirements may be quite stringent (such as a few seconds).

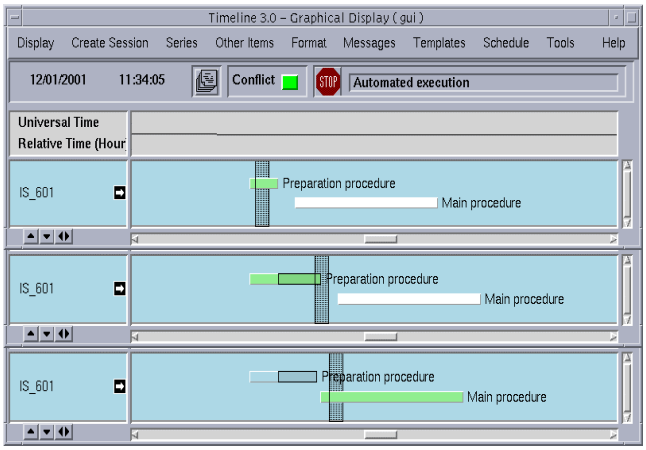

Figure 3: Task duration extension propagation

Top : The Main procedure is scheduled to start after the end of the Preparation procedure.
Middle : The Preparation procedure is extended, the Main procedure is automatically postponed.
Bottom : After the end of the Preparation procedure, the Main procedure is re-scheduled and started.

2.4 Optimization and automation integration

The key operational requirements are reactivity and flexibility. Often a frozen schedule is not compatible with

sending emergency commands or managing equipment failures in the middle of a routine procedure.
Reactivity means the capacity to maintain the schedule consistent with all upcoming events, such as task start, task termination, task duration extension, etc. Consequences of these events must be propagated as soon as they occur, providing a consistent view of the future at any time.
Keeping operational flexibility calls for automated real-time resource management and for allowing schedule to be updated at any time. Main requirements are the following:
· Short-time re-scheduling, triggered asynchronously by new task definition or execution events,
· Immediate scheduling of tasks for emergency execution,
· Automated postponing of tasks waiting for a resource that is used by another task (Figure 3)
· Assisted switching to backup resources in case of failure,
·
. Interruption of tasks for allocating resources to higher priority new tasks
· Manual allocation of resources

2 Integrating Resource Optimization and Operation Automation

The key idea of the proposed approach is to integrate scheduling and schedule execution in the same application in short-loop interaction. We call this application the Control Center Schedule Manager (SM). This approach allows handling all levels of schedule management in distributed space ground segments in a consistent way, from high level scheduling to schedule execution.
Such integration is a technical challenge, because the scheduling process (resource allocation and start time determination), which can take from a fraction of a second to ten seconds or more, must not interfere with external interfaces and the timely execution of tasks and is strongly linked with handling resources failures and manual control of tasks.

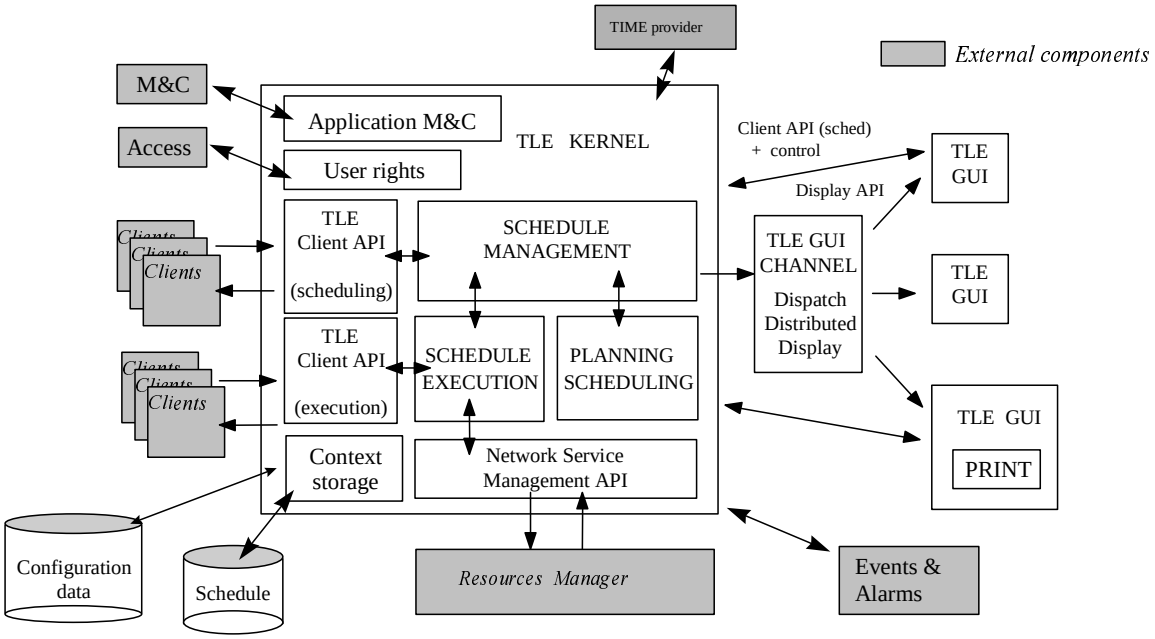

Figure 4: Timeline software architecture

External interfaces, Schedule Management, Execution and Scheduling run in separate threads

3.1 Technical solution

The technical solution relies on the following key features:

1. A multi-threaded software architecture that allows User Interfaces and clients to interact with the schedule and tasks to be monitored even while the schedule is being re-planned.

2. A scheduling scheme enabling to reduce re-scheduling computation time down to a few seconds in case of emergency, and to always keep a consistent schedule, even within very large schedules containing thousands of tasks.

3. A tight and robust coordination scheme between scheduling and execution when scheduling operates near real time or when execution manages resources, through the management of individual software locks on resources and/or tasks.

Coordination between threads

Threads are coordinated along the producer/consumer scheme. The Planning & Scheduling thread reacts on any change on the schedule that requires a scheduling update, such as the creation, edition or deletion of a task, or the modification of any constraint. Such modifications are managed by the Schedule Management thread, but are induced either from the User Interface, from an external application or from the Schedule Execution thread.

Each “schedule change” generates a new “scheduling job” stored in a job stack. Scheduling jobs are ranked along a priority level, depending on the emergency of the change. For instance, a request for immediate execution of an emergency task is ranked at the higher level.

Jobs are processed one by one from the top of the stack, and results are applied to the current schedule. Two main features enable reactivity and efficiency.

1. Compatible jobs are merged to avoid processing several times the same schedule.
2. In case of an emergency job, the current job is aborted (schedule is unchanged), the emergency job is processed immediately and the aborted job re-enters the stack.

Simultaneous scheduling and execution

As Schedule Execution and Scheduling threads rely on the same task parameters (i.e. task start time), their work must be coordinated. On one hand, the scheduling process is not allowed to change dates or resources allocation that will be executed during its own computation time (a maximum over-bound value is used). In other terms, the scheduling process considers that everything that will be in the past when the computation is terminated is frozen.

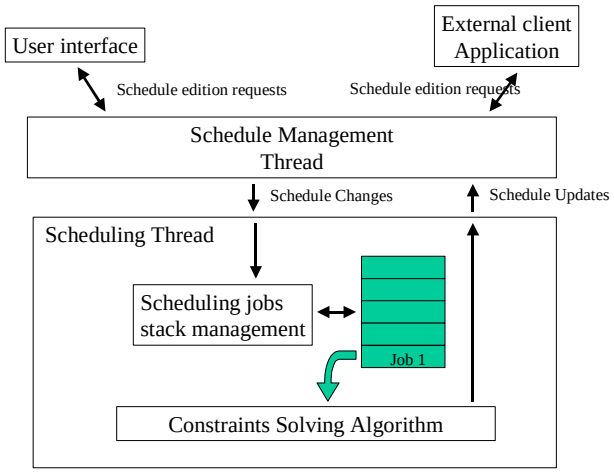

Figure 5: Scheduling jobs are generated according to schedule changes from user requests

Conflict solving and resource allocation algorithm

A dedicated algorithmic approach permits to reach strong reactivity performance, such as requiring only a few seconds for inserting a new task for immediate execution, while supporting a very large schedule capacity.

There is not enough space in this paper to detail the resource allocation and scheduling algorithm. It relies on state-of-the-art constraints propagation algorithms, but its main specific feature is that only a portion of the schedule is (consistently) re-optimized by each “scheduling job”. For instance, in the case of an emergency task requested to start immediately, the algorithm only looks for a free resource (i.e. either not used or used by a lower priority task), but does not modify start times. If the resource has to be taken from another on-going task, this latter is interrupted.

3.2 Operational implementation

The proposed approach is implemented by an operational software product, Timeline, which is openly designed for supervising many kinds of operations. Timeline is part of the Opsware software suite developed by EADS Astrium.

In addition to Schedule Management functions already discussed, Timeline offers essential operational features, such as a distributed and highly configurable User Interface (Figure 6) and open APIs.

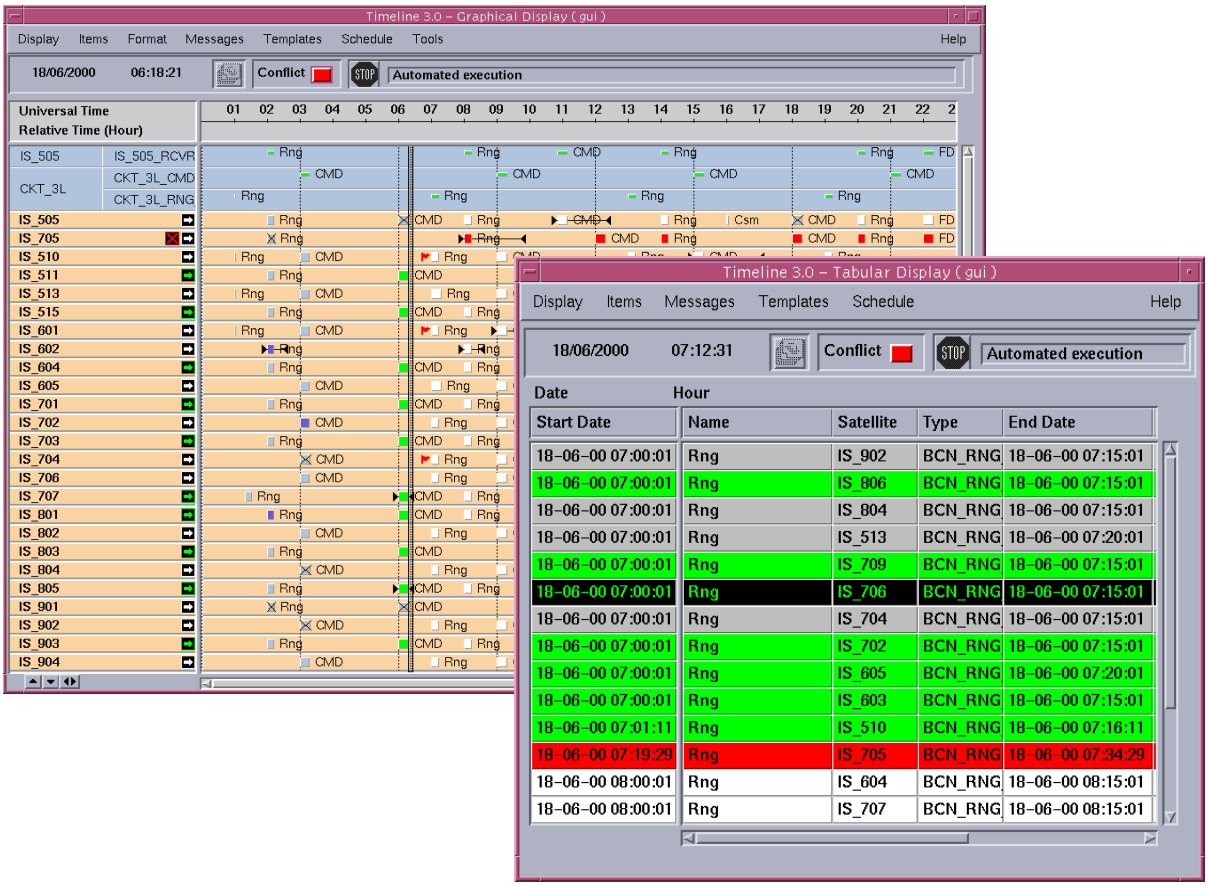

Figure 5: Timeline Graphical Display (INTELSAT configuration, 25 satellites)
Provides a global view of all tasks arranged within a structured and configurable layout.
The Tabular Display provides a chronological list of the current and next activities to start

Timeline also provides high-level schedule editing features relieving operators from routine data typing:

· Tasks are created from templates defining sets of default values and constraints.

· Series of identical activities automatically generated at regular time intervals on a given period of time. This is especially useful for daily operations or routine tasks

. Tasks may be saved in files and then imported and merged within the current schedule. This is particularly relevant for disseminating schedules in distributed architectures (i.e. between a central schedule management system and remote ground stations).

Timeline capacity allows managing schedules containing thousands of tasks, up to 30 satellites and hundreds of individual resources. As an operational application, Timeline runs permanently 24h/24h and supports fault tolerance mechanisms.

3.3 Operational application and benefits

Designed to support various kinds of missions and operations, Timeline is at the heart of the automation of some major operational centers, such as the Helios 2 User Ground Segment, the INTELSAT control center (30 satellites, 50 antennas, 200 resources, 30 days schedule), or the Astrium Control Center.

The first and most obvious benefit comes from an efficient use of resources. For instance, in satellite control centers, TT&C stations are usually sized so that there is one antenna available for each satellite, i.e. one antenna assigned to each satellite. In nominal configuration, there may be conflicts only between tasks for the same satellite that are solved by synchronizing their start times. However, this scheme is disturbed when an antenna has a technical problem or is in maintenance, and the impact becomes critical if communication is lost with a full TT&C station. In this case, the pre-determined assignment must be revised and manual revision is hardly compatible with either emergency or safety. Automated scheduling permits to manage these degraded cases in the same frame as nominal operations.

Short-loop re-scheduling represents significant reduction of working hours as compared to manual planning. Optimization and automation provide together an excellent return on investment as compared to manual rescheduling or a rigid predefined resource allocation.

A second operation benefits comes from the fact that controllers are relieved from routine tasks, such as synchronizing network configuration with commanding procedures or checking for time slots validity, and have a better situational awareness. The third benefit comes from commanding flexibility. When supported at the schedule level, it enables the operators to perform critical, non-nominal operations without having to go back to low-level, manual operations that are prone to human errors.

4 Conclusion

We propose an approach where resource allocation optimization and operations supervision are fully integrated in a single application, the Schedule Manager. Resource optimization then becomes not only compatible with real-time operations safety requirements but also improves efficiency.

Significant real-time resource management automation capabilities include, for instance, switching automatically to a redundant equipment in case of failure, postponing upcoming tasks waiting for resources to be released by a previous one, or even interrupting an on-going procedure to send a emergency command.

These capabilities provide an unprecedented level of automation in Spacecraft Ground Systems managing a several satellites on a commonly shared set of ground equipment and is now flight proven in several major operational ground systems.

Further applications include the Galileo mission schedule management system and ground stations schedule management systems for Space Agencies.

References

J.M. Darroy. and F. Copin, *INTELSAT FDC : A New Era in Satellite Operations. INTELSAT feedback*. Proceedings of the 6th International Symposium SpaceOps 2000, Toulouse, France, June 2000

F. Lecouat and J.M. Darroy, *Tools for Operations Preparation and Automation : the OpsWare Approach*. Proceedings of the 6th International Symposium SpaceOps 2000, Toulouse, France, June 2000

P. Grandjean and F. Lecouat, *Resource Optimization and Automated Operations Supervision: a winning Pair towards Operations Cost Reduction.* Proceedings of the 4th International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations, April 2001

P. Grandjean and F. Lecouat, *Scheduling and Plan Execution: from Ground Segment Automation to Autonomous Spacecraft Operation Concepts*. ESA Workshop on On-Board Autonomy, November 2002

P. Grandjean, *Constellation Mission Planning and Schedule Execution*. Galileo Software Engineering Workshop, October 2003