TIMELINE, A GENERIC TOOL FOR MISSION PLANNING

Yann Parrod, Dominique Corbière and Christophe Rabejac

MATRA MARCONI SPACE, 31 rue des Cosmonautes,31077 Toulouse, France

Fax: +33 62 24 77 80. E-mail: parrod@isis.matra-espace.fr

ABSTRACT.

Reduction of operations costs is a key driver in the development of new Mission Control Centers. MATRA MARCONI SPACE has developed the OPSWARE software package to fulfill this demand and to propose an operational solution for automating operations. This paper focuses on the TIMELINE planning and scheduling component of OPSWARE. It begins with an overview of the OPSWARE command and control components. TIMELINE functions and main features relative to mission planning are described. Future development on distributed planning are then discussed. Finally, application of the tool on current space programs are presented.

1. INTRODUCTION

Space mission operations domain is evolving rapidly. In order to reduce development and operations costs, space operators are now looking for generic tools and are moving to automating operations. Generic tools enable quicker and cheaper development of Mission-specific Control Center (MCC) sub-systems. Operations automation reduces operations costs by relieving MCC controllers from low-level routine tasks during the whole life of the satellite.

The step from manual to automated execution calls for new tools which enable to capture much more information about the mission such as the detailed operations instructions, the constraints for performing these operations and the strategy to react to actual performance of operations. A large part of the workload is transferred from the execution phase to the preparation phase. One has to specify, check and validate activities and schedules before submitting them to automated executers.

To fulfill this demand for an increase in system autonomy, MATRA MARCONI SPACE has developed the OPSWARE package. It is a set of complementary, inter-operable and generic tools which cover all operations phases, respectively preparation, planning, command and control, and performance analysis.

2. OPSWARE

One of the main objectives of OPSWARE is to provide a high level of automation of operations.

The basic principle is to define the procedures using a formal language that supports a syntax close to the natural language used in operations and to execute automatically the procedures using an executer of this language.

OPSWARE provides two levels of automation, respectively the procedure and the plan levels. The OPSWARE architecture which implements these two levels of automation is shown on the figure 1 with the four OPSWARE components: OPSAT, OPSEXECUTER, TIMELINE and TIMELINE-EXECUTER.

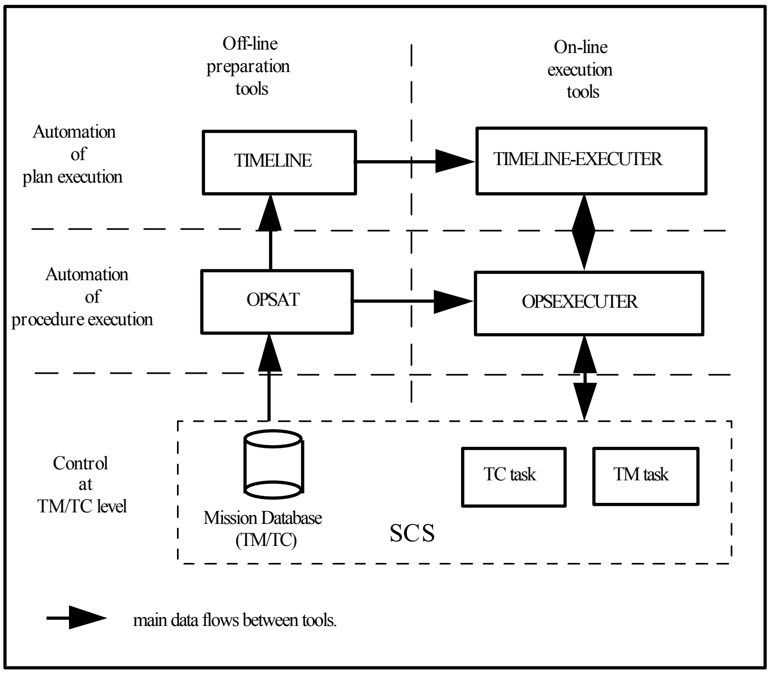

Figure1: OPSWARE components for command and control of operations.

The procedure level is the first level of automation provided by OPSWARE. Flight and ground control procedures are prepared with OPSAT. They are stored in the Procedures database from which they can be accessed by the three other tools.

A procedure is executed automatically or manually with OPSEXECUTER It executes successively each instruction of the procedure and sends the instruction command to the core SCS. OPSEXECUTER enables to manage a stack of time-tagged procedures which can be run concurrently.

The plan level is the second level of automation provided by OPSWARE. Plans of operations are built with TIMELINE. One defines all the mission constraints and TIMELINE computes the feasible execution time windows for performing activities.

The schedule is sent to TIMELINE-EXECUTER for execution. It activates the execution of activities at their scheduled times. A start command is sent to OPSEXECUTER which in return informs the executer on the progress of the activity. The schedule is continuously updated to take into account actual execution times of activities. Re-scheduling may be performed when unforeseen events make the current schedule not anymore feasible.

3. TIMELINE

TIMELINE is a planning tool which enables to define a plan of activities, to schedule them in time, to check if planning constraints are satisfied and to support user in solving conflicts. TIMELINE can be used at all stages of the operations, respectively the preparation, LEOP, routine and final de-orbiting phases.

TIMELINE 2.0 is written in C++ and is available on Unix and Windows platforms.

Objects of the plan

It is possible to create the following objects in a plan:

. Activity. The central objects in a plan are the activities. They are identified by a code which is unique in the plan. They have an expected duration. Their scheduled start and end dates is computed by TIMELINE. Most activities are associated to procedures defined in the procedures database.

. Event. Events can be considered as null duration activities. Typical events are the injections, the apogees and perigees of the satellite.

. State resource. It is a resource which can be in one or several states at the same time. Availability of a state over time is defined by a list of time intervals called segments.

State resources are primarily used to model orbitography information generated by the Flight Dynamics (FD) system. For instance, a state resource is defined for each ground station. The state resource has several states, respectively TM, TC and ranging states which correspond to all types of visibility. Periods of visibility are defined as segments.

Planning constraints

TIMELINE supports several types of constraints on activities and events:

. temporal constraints defined either as time bounds or as precedence links between objects of the plan. Time bound constraints enable to specify an earliest date for the start of an activity and a latest date for the end of an activity. A precedence link enable to define a relative temporal constraint between the start and/or the end of two objects of the plan (activity, event and/or resource segments).

. logical link. A logical link is particular type of precedence link between two activities. The successor activity cannot be executed if the execution of the predecessor activity has failed. This information is used during execution to control the activation of procedure execution.

. resource constraint. An activity may require one or several resource states to be executed. Several types of resource requirements are possible. The activity may require:

- the state of a specific resource, e.g. TM visibility for the Kiruna station.
- a state without any requirement on the resource which provides the state, e.g. TM visibility.
- a backup state, which means that at least two different resources must provide the same state during the duration of the activity execution.

The user interacts directly through a graphical timeline view of the schedule, structured into several domains which reflect distinct areas of activities (e.g. platform operations, payload operations, ground segment activities, etc). The user builds the plan by creating successively new activities and by defining the scheduling constraints. The interaction is entirely based on mouse operations.

Scheduling

The schedule is automatically re-computed in real-time after each user action. Scheduling consists in propagating the temporal constraints and in checking the resource constraints.

The temporal constraints are propagated through the graph of activities defined by the precedence links. The Time Feasibility Windows (TFW) which define the maximum time intervals in which it is possible to schedule each activity are computed and displayed. Activities are all placed at the beginning of their TFW according to an a-soon-as-possible scheduling strategy.

The system controls the edition of the plan so that temporal constraints are always satisfied. It prevents the user from adding or modifying temporal constraints (time bounds and precedence links) which would generate a temporal conflict. For instance it is not possible to move an activity outside its TFW and object attributes which have been specified a conflicting value are set to their previous values.

The resource constraints are checked. Activities requiring a resource state must be scheduled during one of the segments of the state. Activities which are not completely included in a segment are highlighted on the timeline to show the conflict. It is the responsibility of the user to modify the constraints of the plan to remove the conflicts. One may for instance force an activity to be scheduled within a given state segment by defining precedence links from the start of the segment to the start of the activity and from the end of the activity to the end of the segment.

4. DISTINCTIVE FEATURES RELATIVE TO MISSION PLANNING

In this section we stress the functional points that makes TIMELINE specially well suited to fulfill mission planning requirements.

- Interface with a procedures database.

TIMELINE provides on-line access to any operations procedures database supporting SQL protocol. One can insert procedures into the plan and get their main attributes (e.g. duration, definition of parameters). Parameters of the procedures can be instantiated in TIMELINE and stored into the plan with the planned procedures in order to provide a fully instantiated schedule which is ready for execution. It is possible to automatically update procedure attributes when the procedure database has been modified and to clearly identify the procedures which have been modified.

- Modeling mission analysis data.

Data derived from mission analysis can be represented using generic TIMELINE objects. All mission events can be represented as TIMELINE events: geometrical events such as perigee, apogee, ascending and descending nodes, events related to satellite manoeuvres such as injection and start of station-keeping manoeuvre. Orbital states of the satellite can be generically defined using the state resources. For instance, a TM visibility with a ground station is modelised by defining a ground station resource with possibly several states such as TM, TC and Ranging states. A state segment defines a particular ground station visibility of the satellite.

- Interface with Flight Dynamics.

Generic interface is provided to import data generated by mission analysis. It can be used in two ways.

When building a new mission plan, one first imports mission events to fix the time framework of operations. Operations are scheduled and synchronized with respect to these events. For instance, in the LEOP preparation phase, automatic import enables to quickly build several scenarios corresponding to different launch times.

In the execution phase, import from Flight Dynamics can also be used to update the predicted times of the schedule events with the most recent mission analysis data which take into account actual position of the satellite. The schedule is automatically re-computed to assess the impact of these new times.

Mission analysis also computes some of the parameter values of planned procedures, e.g. thrust duration for station keeping manoeuver. These values can also be directly imported in a schedule and stored in the plan with the activity.

- Constraints on the execution of operations.

Precedence links enable to define a sequential order between operations activities. One can associate a delay on the precedence link to synchronize the start time of a procedure with respect to a mission events. State resource constraints enable to specify that a specific state on the configuration of the mission is required to execute an activity, e.g. an operation requires visibility with a ground station or must be performed outside an eclipse period.

- Reporting

It is possible to generate a detail listing of the procedures including the procedure steps and their scheduled time, in a clear table format.

- Inter-operabiliy with other OPSWARE components.

A key feature of TIMELINE is its capability to work in cooperation with a set of complementary tools to provide a global solution to automate operations.

Functional roles and interfaces of OPSWARE components are well defined. Data are transferred by import methods and there is no duplication of information. Same object models, same functions and same MMI are used whenever it is possible, to improve consistency and homogeneity between components and continuity between preparation and execution phases.

TIMELINE generates a schedule which can be directly executed by TIMELINE-EXECUTER. Not only the computed schedule times but all the planning information is transferred to the executer. The executer gets the time margin which is available to perform an activity without violating mission constraints. Mission constraints are also provided. This information allow the executer to adapt the schedule to take into account actual execution events and to perform re-scheduling, that is to search for a new schedule solution when the current one is not anymore feasible.

5. DISTRIBUTED PLANNING

The control of operations in a Mission Control Center is a team work which involves several persons who have different roles. They frequently communicate to exchange data, to share technical expertise or to validate operations.

One may give several examples of interaction:

. When the planning is distributed over different domains which are under different responsibility, planners have to consult each other to build a conflict-free schedule because they share common facilities. Domains can be the payload and the service modules or different satellites.

. The planners build the sequence of operations and in return the controllers inform the planners on the way the operations have actually been performed.

. During the LEOP phase, several technical experts attend operations and have to be clearly informed on the progress of operations.

New groupware technology now enables to propose a new generation of tools to improve significantly the efficiency in group work. Several users can have simultaneously on-line access to the same information, can exchange messages and work concurrently on this shared information.

The next release of TIMELINE will provide an object-oriented client/server architecture based on CORBA to support distributed planning and scheduling.

Several users working on different terminals can be connected at the same time to the TIMELINE server and work on the same plans which are managed by the server. Users can have different views of the same plan at the same time.

Concurrent plan edition will be supported. It will be based on a token mechanism. Only one user is able to edit a plan at a time. This user « takes » the token and the edition of the plan is locked for other users. The server modifies the plan and computes the impact on the schedule. The views over the schedule for the other users are automatically updated to take into account these modifications. When the edition is completed, the token is given back to other users.

6. APPLICATIONS

TIMELINE is commercialised with the OPSWARE package.

It is used on the NILESAT and SINGASAT telecommunication programs which development is under MMS responsibility.

OPSWARE has been selected by ALCATEL for the command-control of the WORLDSTAR satellites and has been selected by SES for its new ASTRA Mission Control Center.

TIMELINE will also be used in the future generic Mission and Control Facility developed by MMS for LEO satellites used on earth observation programs.