Layered Planning Model for Distributed Schedules

Stewart Hall[1] and Roger S. Thompson[2]
SCISYS UK Ltd, Chippenham, UK, SN14 0GB

Mission Planning systems have traditionally focused on the generation of a simple queue of time-tagged telecommands. As on-board autonomy and use of ground automation increases, schedules are increasingly complex and distributed. Schedules initiate procedures and software functions as well as discrete commands and in response to orbital position or event as well as time. From an operational perspective it is important that distributed schedules are consistent and coordinated, and that there is an integrated view of the current status of operations that synthesizes the latest available information relating to all executing schedules. A layered model for Mission Planning and Schedule Execution is proposed that is capable of supporting distributed schedules and integration with automation. The focus is on representing plans and schedules as a set of abstract items describing predicted events, spacecraft contacts, planned operations, schedulable activities and the constraints between them. Operations are tagged by a triggering event, which has an abstract ID but may also be elaborated to absolute or predicted position and time. The definition of a common model of the schedule and its dynamic update allows for development of mission independent schedule displays and schedule history. The model is layered to ensure that planning is performed on a generic view of an integrated schedule, with translation of the schedule to mission specific execution formats encapsulated at the lowest layer of integration with the spacecraft and mission control system.

I. Introduction

THE output of spacecraft Mission Planning systems has traditionally been focused on the production of a mission timeline of time-tagged telecommands. Increasing autonomy and automation within space missions leads to the execution of spacecraft operations being delegated to a distributed set of automated schedules, both on-board and within the ground system. Schedules are increasingly sophisticated: they may reference procedures and software functions as well as discrete commands – and may be triggered by position or event as well as time. It is not unusual for a single spacecraft to provide multiple ways of autonomously executing commands:

- Mission Timeline (MTL) for timed execution of commands
- On-board Position-based Schedule (OPS) for position-based execution of commands, where entries may be flagged for Single (SRC) or Multiple (MRC) Repeat Cycle execution. MRC entries are executed each time the spacecraft returns to the same orbital position, while SRC entries are executed only once.
- On-board Control Procedures (OBCP) initiated by ground-command or via on-board schedules
- Instrument Timelines embedded within individual payloads
- On-board generation of Orbital Events (potentially used to synchronize Instrument Timelines)
- On-board detection of status-based Asynchronous Events
- Event-Action Couplings linking asynchronous or orbital Events to commands or OBCPs.

In addition, there may be distributed schedule automation within the ground segment:

- Within the Mission Control System
- At Ground Stations to enable autonomous support for TT&C and data acquisition passes

From an operational perspective it is imperative that these distributed schedules are consistent and coordinated. A single integrated view of the current status of operations is needed that synthesizes the latest available information relating to all executing schedules, whether on-board or ground-based. This requires the reconstitution of status at the level of items appearing in the schedule – often feedback is only provided by the spacecraft at telecommand level.

---

[1] Consultant, Space Division, SCISYS UK Ltd, Clothier Road, Bristol, BS4 5SS, UK
[2] Technology Manager, Space Division, SCISYS UK Ltd, Methuen Park, Chippenham, Wiltshire, SN14 0GB, UK

Many missions are systematic in nature which leads to the definition of operations in terms of a repeating pattern of operations. Capturing this view of operations at all levels from planning to on-board schedule can simplify the task of mission planning.

SCISYS has developed a layered model for Mission Planning and Schedule Execution that is capable of supporting distributed schedules and integration with automation. This builds on experience from a range of missions, including EUMETSAT's Meteosat 2nd Generation, the GALILEO satellite navigation constellation and the UK's TechDemoSat, and takes into account requirements for future Earth Observation missions. The approach is applicable for use with multiple mission classes in LEO, MEO or GEO orbits.

The focus is on representing plans and schedules as a set of abstract items describing predicted events, spacecraft contacts, planned operations, schedulable activities and the constraints between them, together with the evolving execution status of the schedule. The model is layered to ensure that planning is performed on a generic view of an integrated schedule. Abstracting the planning from the execution domain simplifies the planning and scheduling problem and reduces the impact of changes to spacecraft databases and procedures on the planning operators. The abstract activities are independently expanded into different execution schedules depending on the scheduler protocol e.g. on-board MTL, OPS or ground schedule.

The paper introduces both a Layered Architecture for Mission Planning and automation and a Common Schedule Model used by distributed elements of that architecture.

II. Layered Mission Planning and Automation Architecture

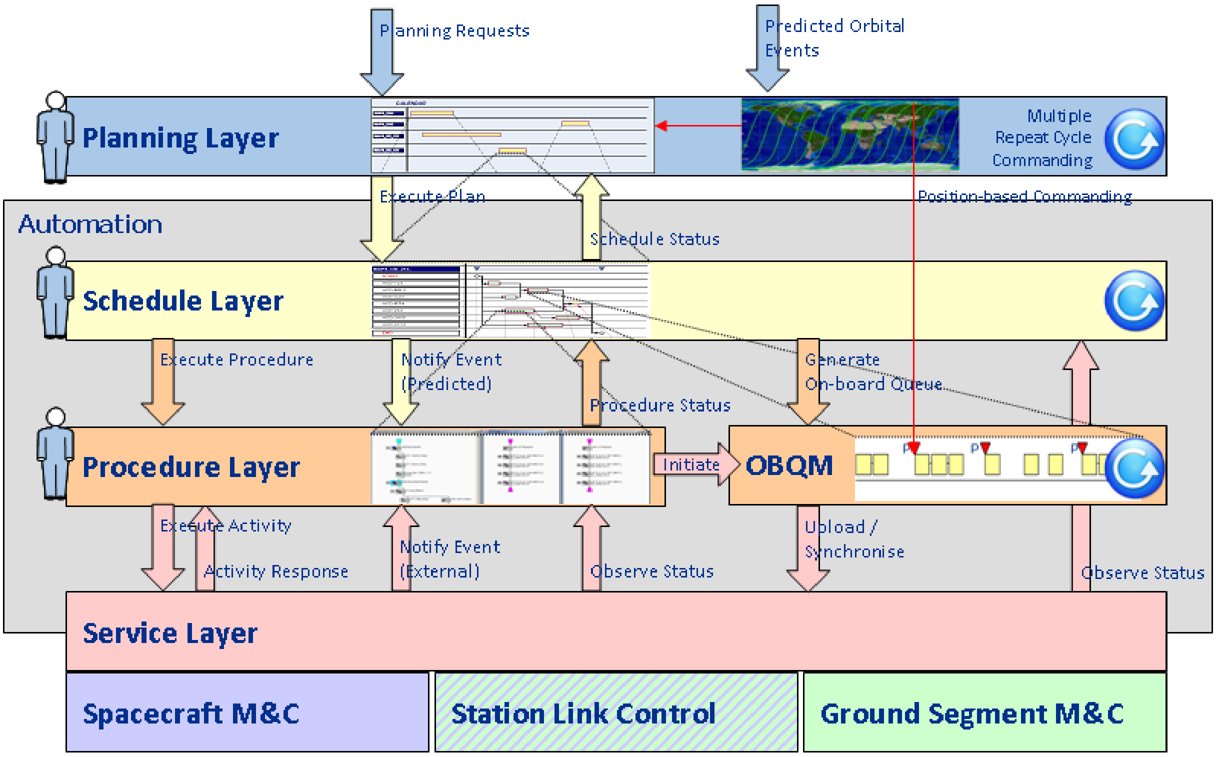

Figure 1. Layered Architecture for Mission Planning and Automation.

Figure 1 illustrates the main elements of a potential layered architecture for Mission Planning, Scheduling and Control within a space mission ground segment. The allocation of functions to components of this architecture is described in the following paragraphs.

A. Mission Planning Layer

The Planning layer at the top of the Fig. 1 corresponds to an off-line Mission Planning System (MPS), which is responsible for the generation of conflict-free schedules of operational tasks (space and ground segment), broken down to individual activities that can be executed either entirely within the ground segment, or entirely on-board a spacecraft. The MPS is responsible for:

- Maintaining the definition of repetitive timetables of routine operations.
- Maintaining standing orders or rules for planning of operations, either periodically or in response to predicted events.
- Selecting the control [TT&C] and data acquisition contacts to be used for mission operations, based on planning rules and ground station availability data. Coordination with external ground network service providers may be necessary as part of this process.
- Provision of tools to support the user entry of ad-hoc planning requests from end-users and the mission operations team.
- Acceptance of ad-hoc planning requests from other functions, including flight dynamics (maneuvers) and on-board software and procedure management.
- Updating the predicted time/position of orbital events based on orbit propagation products provided by flight dynamics.
- Planning of operations requested via timetables, standing orders or ad-hoc requests.
- Generation of a conflict-free schedule for the next period of operations and release of approved schedule changes to the Schedule Execution function.

B. Schedule Execution Layer

The Schedule Layer provides an integrated view of mission operations covering both on-board and ground-based operations. It is responsible for initiating the execution of scheduled operations through ground-based and on-board automation, and for providing an integrated view of current schedule execution status. The Schedule Execution function is responsible for:

- Maintaining the current Operations Schedule, based on updates received from Mission Planning
- Directly initiating ground-based activities at their scheduled execution time
- Providing the set of on-board activities to the On-Board Queue Model (OBQM) function as a set of updates. Identify fixed position repetitive operations that will be implemented as OPS-MRC, such that these are only uplinked when modified.
- Accept status update reports at the level of individual activities from the Activity Execution layer.
- For on-board activities report status in terms of:
    - Upload status (based on confirmation of upload by OBQM)
    - Predicted execution status (based on current time)
    - Confirmed execution status (based on status reports received from the spacecraft)
- Provision of dynamically updated real-time schedule displays to the operations team
- Provision of schedule execution status to the Mission Planning function
- Archiving of schedule execution history
- Provision of schedule history displays

C. Activity Execution Layer

The Activity Execution Layer has two components corresponding to the execution of ground-based and on-board activities respectively.

1. Ground-based Activity [Procedure] Execution

For those missions with limited ground contact, ground-based activities will principally be concerned with the automation of pass-based operations. It is anticipated that routine ground station (TT&C and data acquisition) operations will be essentially autonomous, based on the provision of the contact (pass) schedule and orbit vectors of the satellites. Ground-based procedures running within the mission control centre are responsible for establishing station links for each pass and for uplinking of telecommands as required:

- To initiate uplink of any updates to the on-board schedules (Mission Timeline and Onboard Position-based)
- To perform any defined in-contact immediate commanding activity

The execution status of automated procedures should be returned to the Schedule Execution function in real-time to allow schedule execution status to be updated.

There is potential to execute non-routine contingency recovery procedures from the ground during contact. Ground procedures may also be used to initiate any other ground-based activities, which may potentially include:

- Ground Segment configuration
- Flight Dynamics tasks

- Mission Planning tasks
- Performance Analysis and Reporting tasks

Although in principle any other ground segment systems can also be automated, this may include science data processing, although this is typically data-driven and not directly managed from the schedule.

2. On-board Queue Model

The On-board Queue Model (OBQM) function acts as a proxy for the spacecraft automation functions: Mission Timeline (MTL) and On-board Position-based Schedule (OPS). The existence of multiple queues on-board the spacecraft gives rise to the following options:

- A single function covering all on-board schedules
- Separate functions each dedicated to a single on-board schedule

There are pros and cons to each approach, but the following coordination issues are noted:

- In the event of OPS failure, all on-board operations will need to be scheduled via the MTL. This could be achieved via late-binding of events to position or time within the OBQM itself.
- There may be timing and uplink bandwidth issues that require coordination of the uplink of MTL and OPS updates in order to ensure they reach the spacecraft sufficiently in advance of execution time.

The OBQM function is responsible for:

- Maintaining a model of each on-board schedule. This is based on on-board activity updates received from both the spacecraft and Schedule Layer and includes representation of:
    - activities already loaded to the on-board schedule
    - activities waiting to be loaded
    - changes to already loaded activities required by the latest planning iteration
    - OPS Multiple Repeat Cycle (MRC) entries and their occurrences. Only changes to the MRC entry need to be uplinked to the spacecraft, but status reports relate to each occurrence. It is also suggested that Mission Planning and Schedule Execution may flag an occurrence as “suppressed” and that in this event, the OBQM will need to schedule the uplink of OPS item Deletion and Insertion commands to effectively skip an MRC command for one repeat cycle only.
- Checking consistency between MTL and OPS to ensure spacecraft restrictions are not violated as a result of conflict between commands issued from the two sources. This is primarily a mission planning issue, but as different synchronization methods are used there is potential for conflicts to arise during execution.
- Generate the telecommand sequences required to synchronize the on-board MTL and OPS with the current plan.
- Report uplink status of new or modified on-board activities to the Schedule Layer
- Report execution status of on-board activities when corresponding telemetry data is available.

If supported, on-board procedures will be represented as a discrete item in the queue. Other activities may require expansion to command level based on procedure or macro-command (command sequence) definitions local to the ground segment. It is recommended that this level of expansion is encapsulated within the automation layer to avoid complexity in the interface with Mission Planning.

The command generation step should include the capability to load all on-board activities via the MTL in the event that the OPS is unavailable. This means that OPS activity position tags are translated into timetags on the ground. Potentially, the latest available predicted event data can be used to perform late binding of Events to Time and thus achieve the best possible accuracy in timestamps.

The OBQM is also responsible for reporting execution status of activities in the on-board schedule. Confirmation of execution can only be provided once the corresponding telemetry has been acquired and processed, and may require reconstitution from lower-level command status in order to provide feedback at activity level for schedule displays and mission planning.

D. Service Layer

The complexity of integrating with multiple spacecraft and ground segment M&C functions can be reduced by working through a harmonized service layer. This approach is taken by ESA in integrating automation functions with SCOS-2000 via the Service Management Framework (SMF) and has also been proposed for the future European Ground Segment – Common Core (EGS-CC)[1]. Internationally, CCSDS is defining a suite of standard Mission Operations services[2]. Care must be taken in the way this service layer is defined and implemented in order to:

- Minimize the complexity of operational procedures (some administrative services should be encapsulated by the Procedure Execution layer and not propagated to the user level).
- Harmonize services (e.g. parameter access, commanding) across space and ground functions.
- Minimize the number of specialized services (e.g. to manage the upload of on-board schedules and on-board software).

A key element of the layered Mission Planning architecture outlined above is that a Common Schedule Model is used across Mission Planning, Schedule Execution and On-board Queue Model functions to ensure coherence, extensibility and maintainability of the system over the mission lifetime. The detailed specification of such a Common Schedule Model is an implementation issue for ground segment developers, however it is possible to derive a high-level concept for the model, based on the analysis of requirements for current and future missions.

III. Common Schedule Model

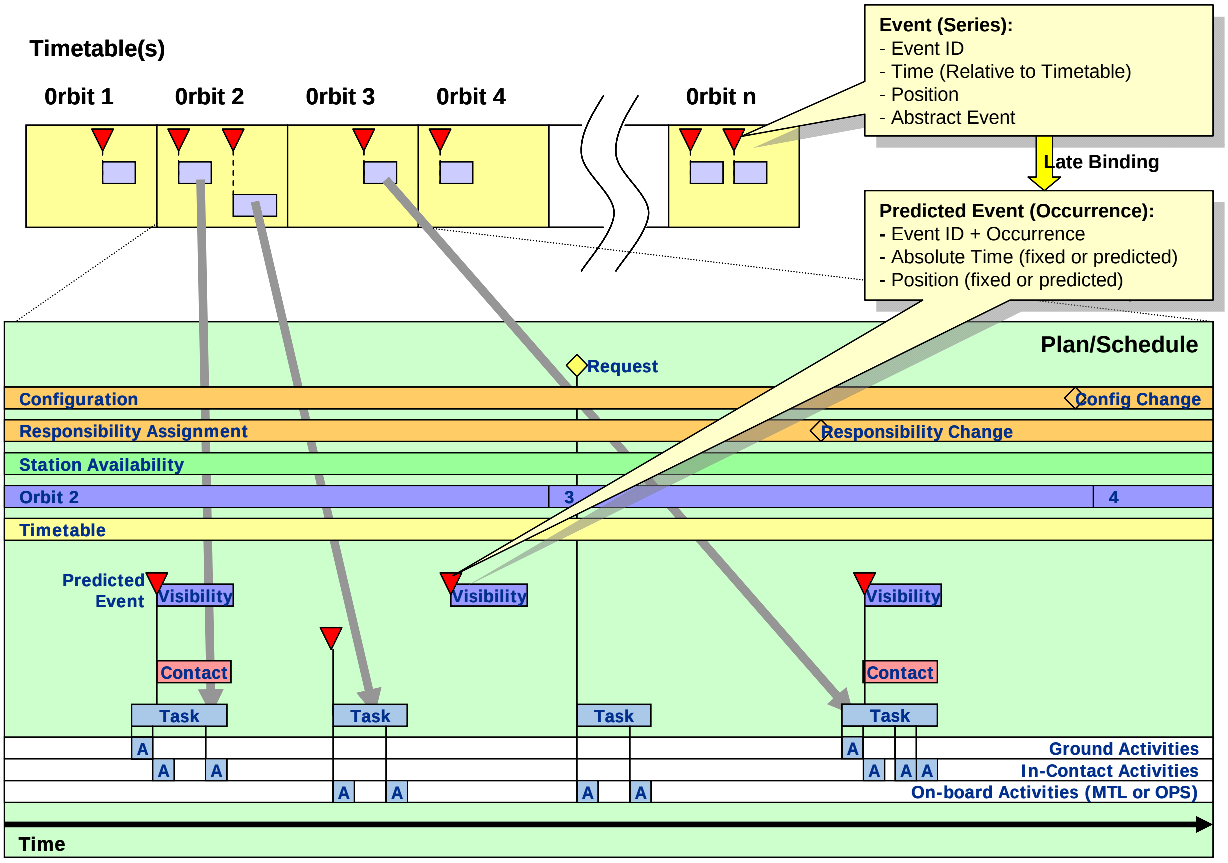

Figure 2. Overview of Schedule Model.

Figure 2 illustrates some of the key features of such a high-level concept.

There are several ways in which operations can be scheduled:

1. Routine operations can be defined in one or more repetitive *timetables*. The *timetable* covers a periodic repeating pattern of operations and can be defined purely in terms of a time period (e.g. daily, weekly, monthly), or more likely for an earth observation mission, in terms of a repeating pattern of orbits. The repeat cycle may be linked to the ground track repeat cycle, but this is not essential. Activities associated with ground station contact are most likely to be defined in terms of a *timetable* correlated with the ground track repeat cycle.
2. Operations may be specifically requested via a *planning request*. Such ad-hoc *planning requests* may be raised by the operations team, or directly by other functions such as Flight Dynamics (maneuvers) or On-board Software Management (software loads).

3. Operations may be placed automatically as a result of a *standing order* or rule, defined in planning configuration data. Typically, this will relate to a ground station contact being planned, or a predicted event being placed in the schedule. Examples are automated scheduling of a pass management ground procedure in response to the scheduling of a TT&C contact; or the scheduling of an on-board operation in response to a predicted sensor blinding event.

Operations defined in Timetables (and usually via the other methods) require the specification of the synchronisation event to be used to initiate them. This may be one of:

- **Time**: if specified in a Timetable this will be a relative time that is resolved to an absolute time when an iteration of the timetable is expanded in the schedule.
- **Position**: this is defined in terms of Orbit number and position in orbit. Such position-based tags can be implemented on-board as OPS-MRC commands if the Timetable is aligned to the ground-track repeat cycle. For monitoring purposes, it is possible to predict the Time of execution using orbit propagation.
- **Orbital Event**: this is an abstract Event that can be resolved by orbit propagation and an occurrence of the predicted event placed on the schedule following generation of an orbit event file. The Position and Time of the Event can then be predicted. This applies, for example, to events that do not occur at the same position on each orbit, such as solar illumination related events (eclipses, terminator transitions, etc.) that occur on every orbit and other events (such as sensor blindings) that occur less frequently.
- **Asynchronous Event**: this is another class of abstract Event that occurs at run-time. Events are raised locally to the schedule execution environment (ground-based or on-board) by other functions in response to observed status. They can be used as a loose-coupling means of synchronising automated operations, e.g. by notifying a threshold transition, Operator intervention or anomalous condition.

Some current and planned future missions have the ability to detect and raise orbital and asynchronous events on-board the spacecraft. Mechanisms for on-board usage of these events are not standardized. They may be used to trigger actions via event-action couplings [e.g. using PUS service 19[3]]; they may be used to synchronize activities within on-board instrument timelines, or it may be possible to reference such events directly within MTL or OPS schedules. It is, however, desirable that an event notification mechanism is supported (both between space and ground segment and within the ground segment) and that this is integrated with the Schedule Execution function.

The use of abstract orbital events within the Mission Planning function and exposing this to the Schedule Model enables late binding to specific scheduling mechanisms based on time or Position. This allows the greatest flexibility in implementing the schedule through on-board mechanisms by encapsulating the mapping to on-board structures as late as the On-board Queue Model (OBQM) function within the mission control system:

- Explicit Times must be mapped to the MTL
- Explicit Positions can be mapped to the OPS. For operations specified in a *timetable*, this can use the on-board Multiple Repeat Cycle (OPS-MRC) mechanism – where the *timetable* itself is essentially loaded on-board. If the OPS is unavailable, then the positions can be translated into times by reference to the propagated orbit and loaded into the MTL.
- Orbital Events can be mapped to Positions based on orbit propagation and then mapped to the OPS using the Single Repeat Cycle (OPS-SRC) mechanism. If the OPS is unavailable, then the positions can be translated into times by reference to the propagated orbit and loaded into the MTL.

Both Positions and Orbital Events would in any case be translated into Times for execution monitoring purposes within the ground segment.

Ideally, the Schedule Model should represent the following information:

- **Configuration**: the version of configuration data applicable to the contained schedule items. This covers the definition of scheduled *events*, *tasks* and *activities*. This could be applicable to the entire schedule, but this is restrictive in terms of managing configuration transitions. It would be better for configuration changes to also appear in the plan, such that forward planning across a configuration change can be accommodated.
- **Responsibility Assignment:** this concerns operational responsibility and can be used to plan the handover of responsibility between different control centers and/or different operational teams.
- **Station Availability**: this represents the availability (or non-availability) of ground stations to support operations and is likely to be an input to Mission Planning. For dedicated ground stations it may indicate down-time for planned maintenance; for other ground stations it may indicate their availability to support mission operations.
- **Orbits**: represents the boundary and naming convention for successive orbits.

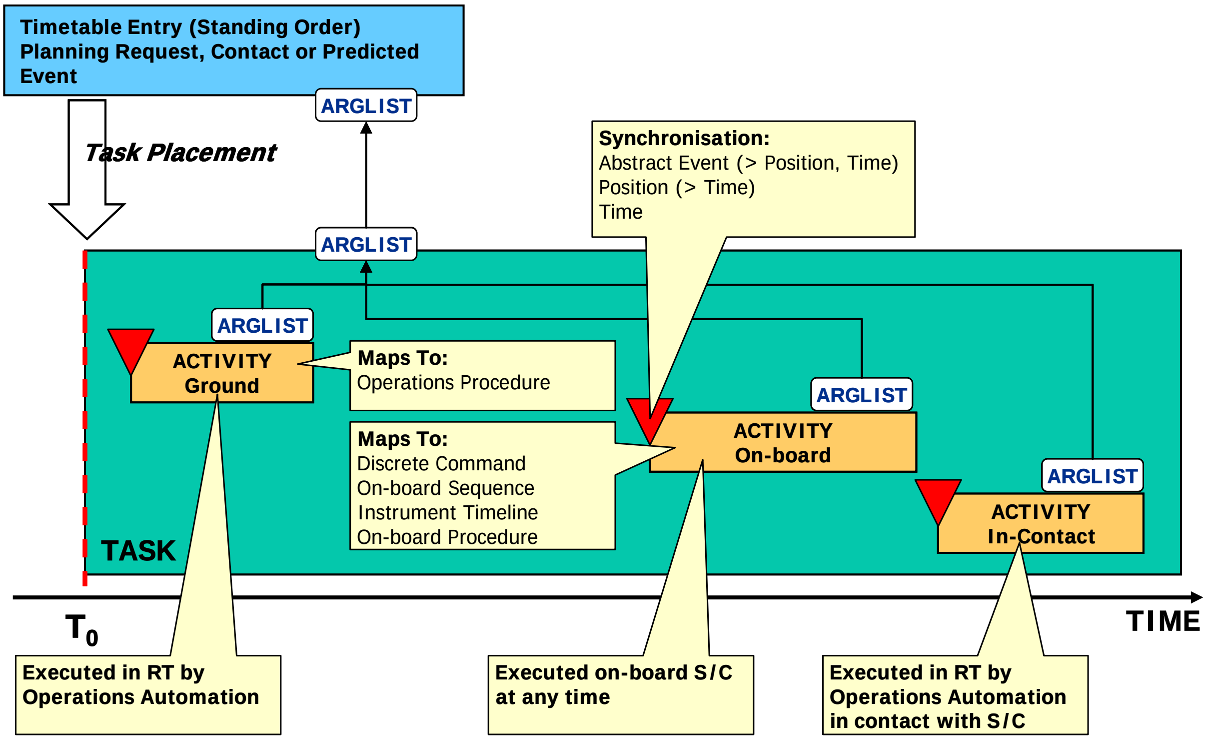

Figure 3. Key Elements of a Scheduled Operation.

- **Timetables**: indicates the repeat cycle (and version) of applicable timetable(s) – the iterations of the timetables will be expanded in the schedule to show individual occurrences.
- **Predicted Orbital Events**: abstract event ID + their predicted position and time. These may be used to initiate or synchronize operations on the schedule. The start-time of the operation is not necessarily at the time of the event, but can be relative to it. Predicted position and time are provided through orbit propagation products and may be periodically updated. It is important when predicted position and time are updated, this is correlated to the Events already in the schedule – it should not be necessary to delete and recreate the events as this will break the links with already scheduled items.
- **Visibilities**: potential ground station contacts based on propagated orbits. Not all visibilities are necessarily used. It is a Mission Planning process to select the visibilities to be used for mission operations. Visibilities can be considered a compound orbital event, comprising two separate events: AOS and LOS.
- **Contacts**: selected ground station contacts for specified purposes (TT&C, Data Acquisition, etc.). Some contacts may be identified as for back-up only. For extended visibilities (e.g. for MEO or deep space missions) only part of a visibility may be used for a contact.
- **Operations**: the operations to be scheduled are expanded within the schedule model. Various detailed models [and terminology] are possible, but it is recommended that there at least two levels:
  - **Task**: a container representing the entire scheduled operation
  - **Activities**: scheduled items that can be individually executed

  Both tasks and their constituent activities can be synchronized with visibilities/contacts and predicted events represented in the schedule model. This can be extended to configuration and responsibility changes.
  Activities must be assigned for execution within a specific execution environment: ground-based or on-board via MTL or OPS. Additionally, activities may be constrained to occur within contact.

Timetables, Planning Requests and Standing Orders all result in the placing of Operations on the schedule. Figure 3 illustrates the characteristics of operations as far as the requirements for representation in the schedule model are concerned.

Timetable entries, ad-hoc Planning Requests, selected Contacts and Predicted Events can result in the placing of a *task* on the schedule.

Definition of the task includes its decomposition into one or more executable *activities*. Other configuration information such as constraints and resource usage may be needed by Mission Planning to support the planning algorithm, but most of this does not need to be exposed to Schedule Execution function. Attributes of tasks and activities that are needed to represent and monitor the execution of the schedule should be included in the model, these include:

- Source or parent object (what caused it to be scheduled)
- Links to other schedule objects (e.g. predicted event, precedent activity)
- Arguments (parameters passed into the *task* or *activity*) and specific to the corresponding *task* or *activity* class definition
- Requested or preferred execution Time, Position or Event
- Planned execution Time, Position or Event
- Predicted execution Time and Position where relevant.
- Expected duration
- Execution status
- Actual execution Time

Arguments or attributes of the source object can be passed through to the *task*, and from there to constituent *activities*. For some types of Operation, there may be a need to reference the attributes or arguments of the preceding occurrence of the same type of Operation.

For *activities*, the definition should also include:

- Identification of where the Activity should be executed:
  - On-ground
  - On-board (via MTL or OPS)
- Whether the Activity must be performed during a Contact of a specific type
- Mapping of the Activity to an executable item:
  - On-ground this may be:
    - An Automated Operations Procedure
    - A Manual Operation
  - On-board this may be:
    - A Discrete Telecommand
    - A macro-command [Telecommand Sequence or Procedure] to be expanded on ground prior to uplink to the spacecraft.
    - An Instrument Timeline implemented On-board
    - An On-board Control Procedure

The above focuses on static aspects of the schedule model, but the dynamic aspects of the model provide a complete view of the evolution of the schedule of operations. Following a periodic Mission Planning session, any updates to the Schedule are released for execution. The integrated (space and ground) schedule is passed to the Schedule Execution layer, together with an indication of whether individual schedule items are:

- New (did not appear in previous schedule)
- Modified (existed in previous schedule but attributes have changed)
- Deleted (existed in previous schedule but has been removed)
- Unchanged (from the previous iteration of the schedule)
- Suppressed (for OPS-MRC occurrences only)

Similarly execution status feedback is provided by the Schedule Execution layer to Mission Planning. This is done at the level of individual schedule items (Tasks and Activities). As a minimum, execution status is reported in terms of:

- Status: Pending, Uplinked (for on-board activities), Executing, Completed
- Actual Execution Time and Duration
- Confirmed Success or Failure
- Reason for Failure

This dynamic schedule model can be used to provide distributed access to current Schedule Status from any function that requires it, including:

- Schedule Execution Displays (various views based on time or position)
- Mission Planning

By recording Schedule History in this same format, it is possible to reconstruct the schedule execution status at any point in the past. This would most probably be in terms of a "final state" view of the historical schedule, although the event based reporting of individual changes to the current schedule would mean that dynamic replay is also possible.

IV. Conclusions

A layered architecture for Mission Planning and Automation has been described, together with a schedule model that provides a common view of the schedule across multiple elements of the ground segment. Adoption of this approach brings a number of benefits for the design of mission operations systems:

1. It allows looser coupling between Mission Planning, Schedule Execution and the ground proxy for on-board schedules (the On-Board Queue Model). This in turn is an enabler for re-use of software components across multiple missions.
2. It encapsulates mission specific features of on-board (and ground-based) schedules in the corresponding proxy function, such that generic Mission Planning and Schedule Execution functions can be used with minimum customization.
3. It supports a generic approach to the triggering of operations by Events associated with absolute or predicted positions and times. This allows for late binding of position- and time-tags based on event predictions from orbit propagation It also supports time and position-based scheduling within a common framework.
4. It allows for the representation of the repetitive operations typical of systematic missions as repeating timetables based on both time and orbital repeat cycles (position).
5. It supports the coordination of schedules across multiple distributed execution platforms as well as synthesizing their execution status in a single overview of the mission schedule.
6. The dynamic schedule update and status reporting model supports distributed schedule displays and recording of schedule history.

Appendix A
Acronym List

AOS / LOS	Acquisition / Loss of Signal
GEO	Geostationary Orbit
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
MPS	Mission Planning System
MRC	Multiple Repeat Cycle (OPS scheduling Mode)
MTL	Mission Timeline [on-board time-based schedule]
OBCP	On-Board Control Procedure
OBQM	On-Board Queue Model
OPS	On-board Position-based Schedule
RT	Real time
S/C	Spacecraft
SRC	Single Repeat Cycle (OPS scheduling Mode)
TT&C	Tracking, Telemetry and Commanding

References

[1]Peccia, N. et al., "The European Ground Systems – CommonCore (EGS-CC) Initiative", *GSAW 2012 On-line Proceedings*, February 2012, http://csse.usc.edu/gsaw/gsaw2012/s3/peccia.pdf

[2]Mission Operations Services Concept, Informational Report (Green Book), CCSDS 520.0-G-3, December 2010, http://public.ccsds.org/publications/archive/520x0g3.pdf

[3]Ground Systems and Operations – Telemetry and Telecommand Packet Utilization Standard, ECSS, ECSS-E-70-41A, January 2003