

Loosening the reins on Sentinel-3 on-board automation
T. Francisco^{a*}, I. Rourich^b, L. Ventimiglia^c, A. Johnstone^d, E. Trollope^e

^a *Flight Operations Division, EUMETSAT, Germany, tiago.francisco@eumetsat.int*

^b *Satellite and Service Operations, Telespazio Germany, Germany, Isabel.GonzalezRourich@external.eumetsat.int*

^c *Flight Operations Division, EUMETSAT, Germany, luca.ventimiglia@eumetsat.int*

^d *Exostaff GmbH, davidandrew.johnstone@external.eumetsat.int*

^e *Flight Operations Division, EUMETSAT, Germany, ed.trollope@eumetsat.int*

* Corresponding Author

Abstract

Missions, such as Copernicus Sentinel-3 which are operated by EUMETSAT, rely on the automation of their satellites to fulfil their goals. While these satellites offer a great deal of autonomy flexibility, the original autonomy concept is rigid in how this flexibility can be used. This is due to the autonomy implementation being split into three distinct categories: FDIR; autonomous mode transitions; and time-based/geolocated operations. These automation categories make use of different PUS services which are not utilised to their absolute potential. These spacecraft also support the use of OBCPs, which are essentially control procedures in a script format that reside on-board the spacecraft, but their use can be restricted due to either on-board constraints or to limitations on-ground such as unfamiliarity with the coding language or lack of suitable development tools. The EUMETSAT Copernicus Sentinel-3 Flight Control Team (FCT) have been exploring on how improve the mission automation by making extensive use of the flexibility provided by the available PUS services and augmenting said automation with OBCPs whenever possible.

Keywords: Automation, Copernicus, EUMETSAT, Mission Evolution, Satellite Operations, Packet Utilisation Standard

Acronyms/Abbreviations

AOCS	Attitude and Orbit Control Subsystem
CDE	Cooling Drive Electronics
CSW	Central Software
FCT	Flight Control Team
FDIR	Failure Detection Isolation and Recovery
FMON	Functional Monitoring
GNSS	Global Navigation Satellite System
MHSTR	Multi head Star-tracker
MTL	Mission Timeline
PMON	Parameter Monitoring
PUS	Packet utilisation standard
OBCP	Onboard control procedure
OPSO	Orbit Position Schedule
S/C	Spacecraft
SLSTR	Sea & Land Surface Temperature Radiometer
SRAL	SAR Radar Altimeter
SSID	Sub-schedule ID
TC	Telecommand
TM	Telemetry

1. Introduction

The current Sentinel 3 constellation is comprised of Sentinel-3A launched on 16 February 2016 and Sentinel-3B launched on 25 April 2018, both flying in a sun-synchronous orbit with an altitude of roughly 815km. The constellation is part of the Copernicus programme, and it is tasked with the measurement of the Earth's oceans, ice, land and atmosphere for forecasting systems, environment and climate monitoring. Being part of the Copernicus programme requires the constellation to provide a near-real time delivery of uninterrupted science and measured data from its set of optical and topography instruments. To achieve this the constellation relies on its implementation of the PUS spacecraft services and on-board automation. The Sentinel-3 spacecraft (S/C) offer a baseline automation concept that ensures spacecraft safety as well as reliable, repeatable and predictable autonomous mode changes and operations. It is also possible to create on-board control procedures (OBCPs) that reside on-board and that can be executed whenever deemed necessary. An overview of the Sentinel-3 mission can be found in [1].

Over the constellation lifetime, the Sentinel-3 Flight Control Team (FCT) has been improving the baseline automation concept within its boundaries [2], however the team felt that these boundaries were too restrictive and holding back the real automation potential of the Sentinel-3 S/C. While the use of OBCPs would mitigate the restrictiveness of the baseline automation concept, its use is not without its problems and limitations, which lead the Sentinel-3 Flight Control Team to explore other avenues to achieve a high level of autonomy by utilising the available PUS services of the Sentinel-3 S/C. Several other missions have also successfully exploited the available PUS services to perform operations for which these services were not designed to do, [3], [4] and [5].

2. Baseline Sentinel-3 Automation Operational Concept

The baseline Sentinel-3 Automation Operational Concept is split between three distinct categories that do not overlap with each other. These categories make use of the implementation of the PUS-A standard [6], [7] available in Sentinel-3 S/C to achieve their desired autonomy, and this use is also quite distinct, with each category only using a subset of the available services and not using services that were “designed” for other categories.

The baseline automation concept allows for the addition of new automation solutions if they fit into one of the distinct categories and don't deviate from their paradigms.

There is also the internal autonomy of several instruments on-board the Sentinel-3 S/C, for which the flight control team as the means to monitor, manage and configure the automation parameters. However, the Sentinel-3 EUMETSAT FCT does not have visibility of how this autonomy is created as this is provided by the instrument software themselves, software which is provided by the manufacturer of the instrument.

2.1 Baseline Automation Concept Categories

Automatic instrument mode changes

The Sentinel-3 S/C are capable of autonomously changing the mode of any instrument based on certain criteria. This is possible in a variety of ways [2], however the only implementation of this in the baseline automation concept are transitions triggered by S/C events.

Spacecraft safety / FDIR

The concept of Failure Detection Isolation and Recovery (FDIR), ensures spacecraft safety by autonomous monitoring of telemetry and executing autonomous safing actions whenever the monitoring system detects one or more violations of safety defined limits, or whenever certain events are generated by any unit on the Spacecraft. The safing actions depend on the triggering unit. Payloads are usually switched off whereas platform units trigger their redundancy. Depending on the number and combination of platform safing occurrences, the S/C may end up in safe mode.

Several units also offer their own internal safety mechanisms that can be augmented by combining them with the overall Spacecraft safety mechanisms. One example of this is how the Sentinel-3 MHSTR can reboot themselves when their software detects a mathematical error and the Sentinel-3 S/C can swap to the ground selected redundant unit if this reboot was unsuccessful in addressing the issue.

Time-based or Geolocated Operations

In addition to the time-based commanding that the Sentinel-3 S/C offer, there is also the capability of executing commands over a specific position on any orbit over the mission defined repeat cycle, which is 384 orbits over 27 days. It is mostly used for routine operations that are position based such as the SRAL instrument operations. Further details on the instrument can be found in [8].

2.2 Baseline Automation Concept building blocks

Parameter monitoring (Service 12)

The Sentinel-3 S/C implementation of the PUS Service 12 allows the of monitoring of parameter values against a set of low and high limits or against an expected value. It is possible to define a validity condition for the monitoring in question as well as defining the frequency of the monitoring check and the amount of out of limits/unexpected number of samples needed to trigger the monitoring. If triggered the monitoring will generate an event that is defined by the flight control team.

Event action (Service 19)

The Sentinel-3 S/C implementation of the PUS Service 19 allows the S/C to execute an action (telecommand, action sequence start or OBCP start) whenever an identified event is generated by any unit of the S/C.

Functional monitoring (Service 142)

The Sentinel-3 S/C implements the custom PUS Service 142 which allow the monitoring and management of several PMON (S12) by grouping them into unique groups called Functional Monitoring, or FMON (S142) for short. It is possible to add a validity condition to the created FMON (S142) groups. The FMON (S142) group will take care of enabling and disabling its monitored monitorings as well as generating a ground defined event if any of the grouped PMON (S12) trigger. In the PUS-C [9] this functionality is covered by PUS-C Service 8.

Mission timeline (Service 11)

The Sentinel-3 S/C implementation of the PUS Service 11 allows the S/C to execute commands that upon the S/C reaching the designated time. The service allows the manipulation of the “time tagged” commands and allows to select a sub-schedule identifier (SSID) for each command such that bulk operations can be performed.

Orbit Position Schedule (Service 131)

The Sentinel-3 S/C implements the custom PUS Service 131 which allow the S/C to execute commands that are “geo-tagged” and that are to be executed once or every time the S/C flies over the desired position. The service allows the manipulation of the “geo-tagged” commands and allows to select a sub-schedule identifier (SSID) for each command such that bulk operations can be performed. In the PUS-C [9] this functionality is covered by PUS-C Service 22.

Action sequences (Service 132)

The Sentinel-3 S/C implements the custom PUS Service 132 which allow the S/C to perform a sequence of commands with configurable deltas in-between. These sequences reside in the safeguard memory of the S/C and need to be coded by ground with the aid of a special tool. In the case of the Sentinel-3 mission, these must be provided by the S/C manufacturer. In the PUS-C [9] this functionality is covered by PUS-C Service 21.

2.3 Baseline Automation use cases

Spacecraft Safety (FDIR)

Except for the internal Sentinel-3 S/C CSW watchdogs, all of the S/C wide FDIR is handled by using a combination of the above building blocks. For example, the health of both the nominal and redundant GNSS units are monitored by the S/C. This is achieved by each unit having its own FMON (S142) that is monitoring several PMON (S12) of said units. Should a PMON (S12) trigger, the FMON (S142) where this PMON (S12) resides triggers and then this causes an action to be executed. The next diagram illustrates how the GNSS units are monitoring on-board.

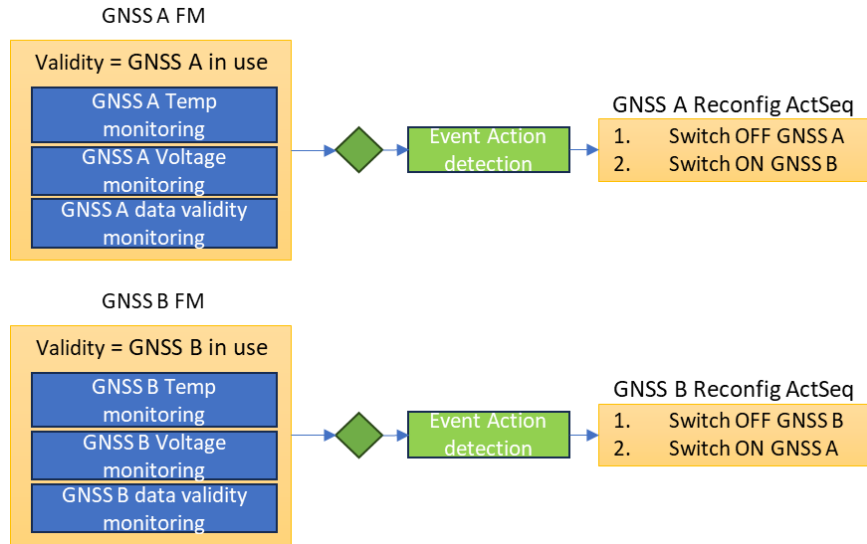


Fig. 1. GNSS FDIR implementation

Automatic instrument mode changes

Several instruments on-board the Sentinel-3 S/C have their mode automatically changed based on whether the S/C is on the day or night side of the orbit. This is done by adding an entry to the of the event action service such that, the appropriate transition is commanded whenever the day or night transition events are generated [2].

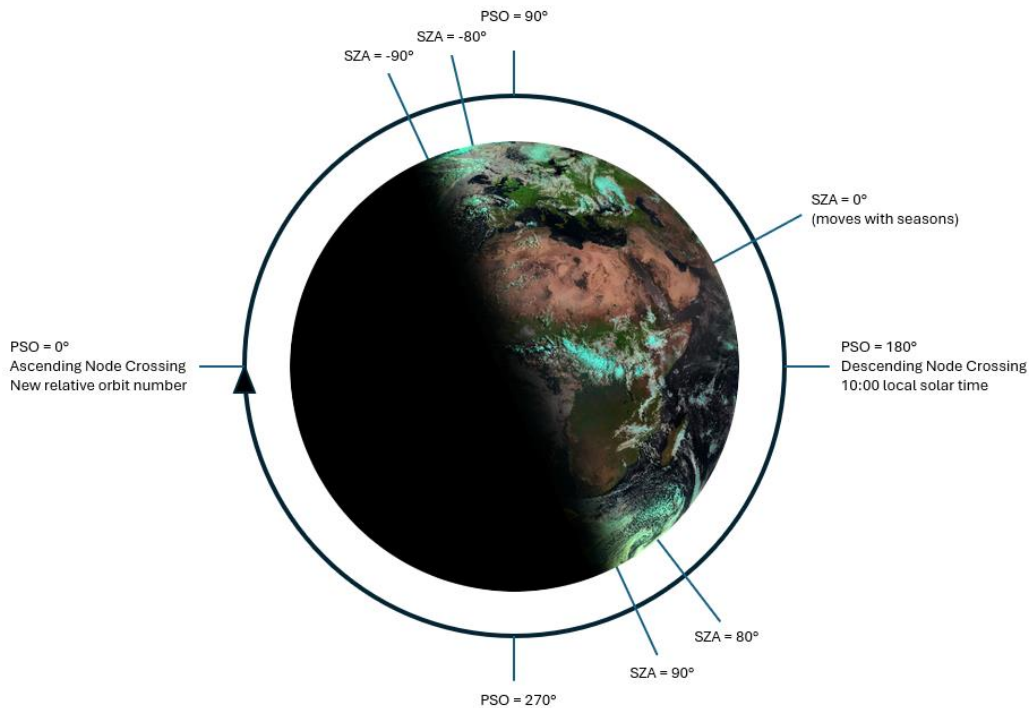


Fig. 2 Diagram showing Orbit Position (PSO - Position Sur l'Orbite) and Sun Zenith Angle (SZA) as used on Sentinel 3

Time-based or Geolocated operations

The Sentinel-3 S/C allow the possibility of having commands onboard that will be executed either when a selected time is reached or based on the spacecraft geolocation. The daily communication passes are possible due to the carrier on and off times being time tagged onboard and the dump of science that is geolocated over the ESA north pole ground stations. The SRAL instrument [8], which is a radar used to measure topography, is commanded between its two different tracking modes via the commands are based on the spacecraft geolocation.

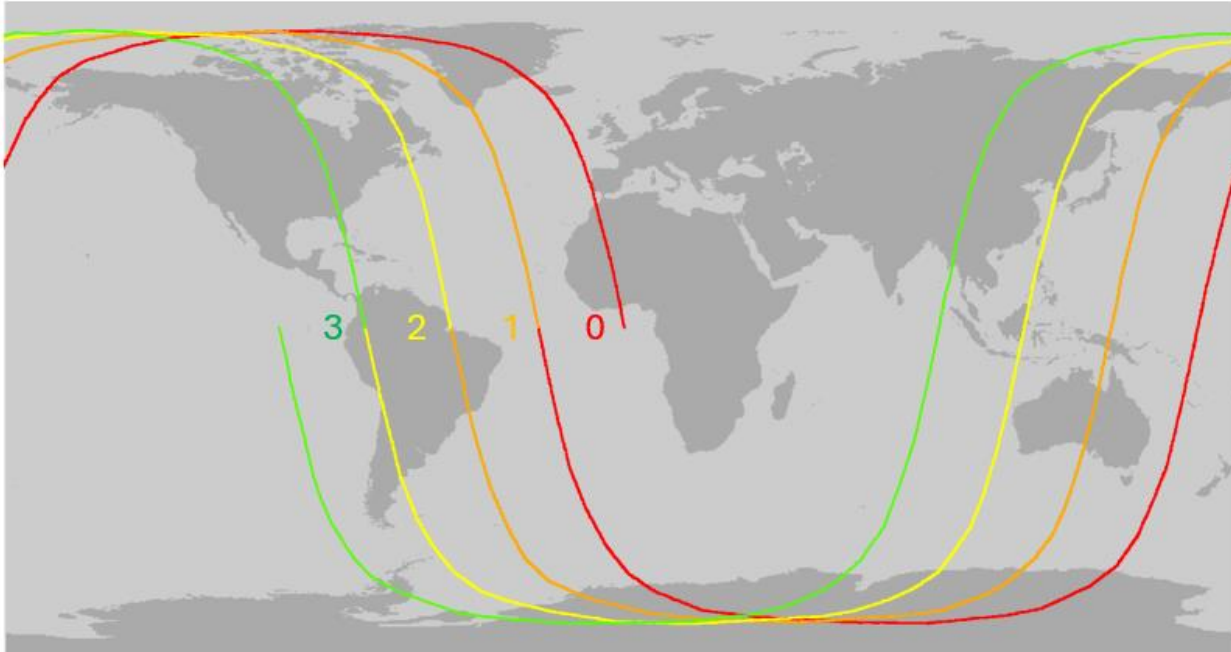


Fig. 3 Diagram showing the first four Repeat Orbit Numbers (RON) of a 27-day, 385 orbit repeat cycle

2.4 Baseline Automation limitations and shortcoming

The offered solutions are quite robust and can be expanded on by adding more of the same types of automation operations. But this is where the restrictiveness of the concept begins to appear as the very design of the solutions implemented do not fully exploit the building blocks that it uses. In other words, the offered automation implementation does not go beyond the main categories, even though the “building blocks” themselves can be used for much more than that [2].

While some instruments offer internal automation capabilities, this differs from instrument to instrument and is something that can't really be manipulated by the flight control team, besides the usual management.

Baseline automation operational concept limitations

- In the baseline automation concept, one occurrence, be it an out of limit of an event generation, only triggers one action.
- Action sequences cannot be maintained nor improved by the Sentinel-3 EUMETSAT FCT
- While it is possible to apply masks to the values monitored by the PMON (S12), it is still not trivial to monitor a parameter for an exact unique value and have that value trigger the PMON (S12)
- Functional monitoring only used to group parameter monitoring of same unit
- Not possible to perform an action sometime after an out of limit triggered

Shortcomings

- Heavy reliance on action sequences, that the flight control team cannot maintain nor develop new ones
- Not scalable
- FDIR limited to switch off, reconfigure or safe mode
- Automatic mode change without any logic behind it
- Repeatable operations that are triggered by certain criteria not present in automation concept
- Assumption that any automation needs will be tackle by OBCPs.

3. Sentinel-3 OBCP Use

To allow the Sentinel-3 flight control team the capability of expanding the on-board automation beyond what is offered by the baseline automation concept, the Sentinel-3 S/C allow the use of on-board flight control procedures (OBCP).

These are “scripts” that reside in the on-board memory and can be activated by the Sentinel-3 S/C by telecommand whenever the FCT deems it necessary. This can be by direct commanding or by tagging it to a certain criterion (event, out of limit, time or position tagged).

A dedicated tool is necessary to create the logic behind the OBCP and then a compiler is needed to convert the desired logic into a format that the on-board OBCP process can interpret. These tools are provided to the FCT by the S/C manufacturer, which is also responsible for the maintenance of said tools.

The use of OBCPs in the Sentinel-3 mission is ramping up due to the following points:

- Possibility to create on-board logic and mathematical checks
- Possibility to self-contained operations with various branches
- Possibility to “replicate” action sequences without the need for support of the S/C manufacturer

The use of OBCP allows the S/C operators to greatly augment the autonomous capabilities of the any S/C that allows their use, however there are several unique reservations to the use of OBCP in the Sentinel-3 S/C on top of the already existing limitation of the OBCP system:

- There are only 20 slots available for OBCP storage in the Sentinel-3 S/C
- It is only possible to have 3 concurrent running OBCPs
- Due to a limitation in the Sentinel-3 CSW, there is the possibility of an OBCP becoming “frozen” which would block one of the 3 available slots until it is aborted by the FCT via ground commanding.
- There is a 5000 byte class code limit on each OBCP. For reference, the default OBCP that takes care of initiating the global dumps for the Sentinel-3 S/C has a class code with roughly 4000 bytes.
- Commands from OBCP have lowest priority in the CSW processor resource allocation. In other words, all other commands in the same processor cycle will execute before any OBCP commands can execute. For comparison, commands originating from an event action (Service 19) have the second highest priority

There are also other important aspects to keep in mind when deciding to use OBCPs to perform on-board automation

- Since the OBCP software tools must be provided by the S/C manufacturer, there might be a time where these tools are no longer supported and receive no further updates
- There needs to be a knowledge transfer between team members that might not be proficient with coding practices
- Discipline when creating OBCPs is needed such that the 5000 byte size limitation is not exceeded
- While minimal, training beyond usual S/C operations is still required
- OBCPS need to be maintained, and version controlled

Due to the limitations described above, the Sentinel-3 EUMETSAT FCT will mostly rely on OBCPs to augment operations and other automation solutions. Special effort will be put in trying to create OBCPS that are multi-function to address the limited number of slots available in the system. Fully automated OBCP solutions will be considered but not be the first choice.

One example of the use of OBCPS that the Sentinel-3 EUMETSAT FCT is creating to augment existing operations is an OBCP that generated custom events whenever needed:

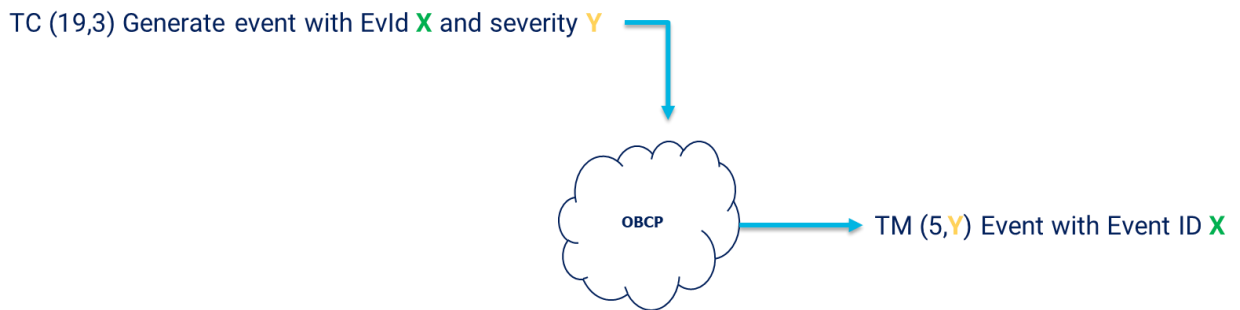


Fig. 4 Custom event generator

4. Evolution of Sentinel-3 Automation Operational Concept

As alluded in the beginning of this paper, the Sentinel-3 EUMETSAT FCT has made strides [2] into changing the baseline automation operational concept into something that is more flexible and not tied to specific categories. This is achieved by fully exploiting the capabilities of the available PUS Services and going beyond what these services were designed to do.

In essence, with this new approach to automation, the previous categories can be abolished and instead of relying on trying to fit specific operations into one of these categories, the paradigm is now to look at a automation needs and use the available “toolset” or “building blocks” to achieve it.

To achieve this change in paradigm, the Sentinel-3 EUMETSAT FCT explored new ways to use the available PUS services, especially when compared to how these services were being used in the baseline concept.

Some of the following concepts are already available in the PUS-C standard [9].

4.1 New Automation Concept building blocks

Parameter monitoring (Service 12) evolution

In addition to the already existing within limit checking and value as expected check, it is possible to use the Sentinel-3 PUS Service 12 implementation to perform the following:

- Action timer:
 - PMON (S12) that is always “failed”, thus serving as a timer. Once timer elapses, and event is generated and this event can be used to trigger one action
- Expected value monitoring:
 - By default, if a PMON (S12) entry is created with an “expected value”, the entry will trigger if the parameter value is any value except for the “expected value”. In order to have a PMON (S12) entry trigger when a value matches the “expected value”, then is necessary to swap the monitored parameter with the validity parameter upon creation. Due to the how PMON (S12) works, this check might become instant and not dependant any timer/repetition of values

Functional monitoring (Service 142) evolution

The monitoring and management of several PMON (S12) by a FMON (S142) entry can be exploited to allow to nest actions as well as use these FM as steps in automatic operations:

- Action nesting / Multiple actions from single OOL

- It is possible to group several PMON (S12) into one FMON (S142) and have the corresponding actions execute in order in the same cycle, because the PMON (S12) checks and corresponding event generation are performed in order of their creation ID. Thus, it will allow for the execution of multiple actions from a single entry point, emulating action sequences.
- It is also possible to have several PMON (S12) in a FMON (S142) execute at different interval while maintaining the “action nesting” sequence, but the order changes. The PMON (S12) with the shortest “timeout” will trigger in order, followed by the event generate by the Functional monitoring itself and then any other PMON (S12) with longer “timeouts”

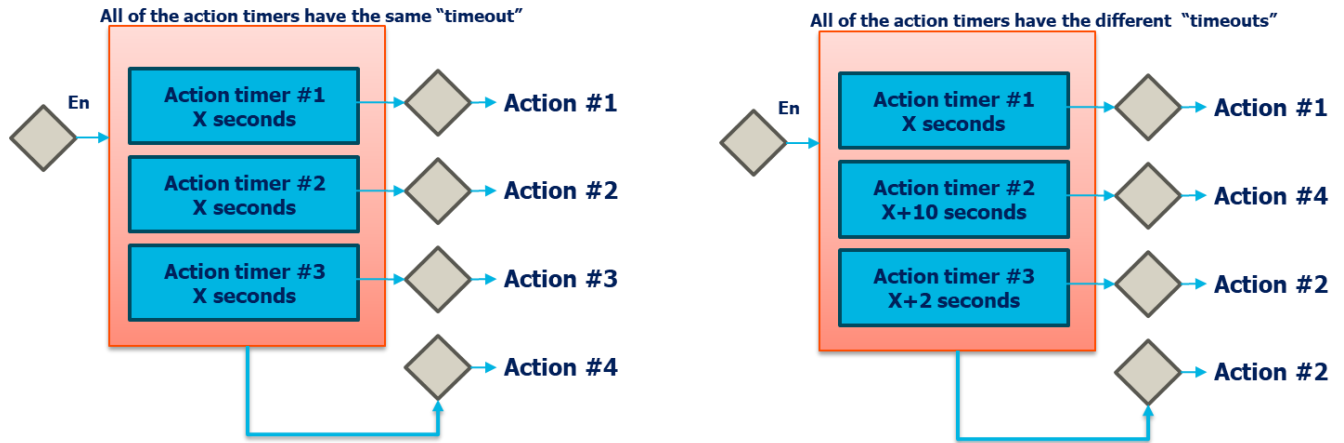


Fig. 5. Multiple actions from single entry point

- Step based automatic operations
 - By chaining the above evolution, it is possible to automatic operations onboard purely running on PMON (S12), FMON and Event Action (S19) . It is also possible to add “logic” gates in between steps, or any other “building block”

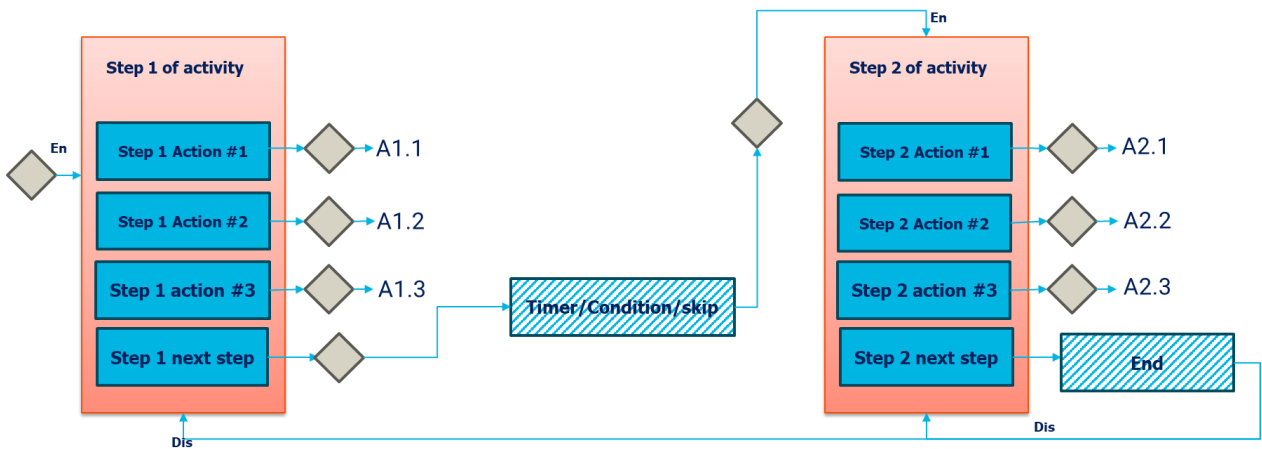


Fig. 6. Step based automatic operations

Orbit Position Schedule (Service 131) evolution

By making use of the OPSO (S131) command manipulation capabilities, it is possible to have permanent operations on-board that have their execution window dynamically adjusted based on a ground defined criterion. To achieve the desired dynamic adjusting, it is necessary to have blocks of command as follows:

- “Push” action block: This block will have a unique identifier (SSID) and whenever it is executed, will shift all blocks forward in orbital position. The position of this block needs to be before the “permanent” operation block.
- “Permanent” operation block: This block will have a unique identifier (SSID).
- “Pull” action block: This block will have a unique identifier (SSID) and whenever it is executed, will shift all blocks forward in orbital position to after the end of the “pull” block.

The criteria to “activate” the shifting is up to the FCT and can be defined via the available automation toolset or even by an OBCP. Once the criteria is defined, it is necessary to tie it to an event action such that whenever the criteria is met, the event action disables the “pusher” block thus allowing the “permanent” activity to take place followed by the “puller” block that shifts the activity to the next execution window. If the criteria is not met, then the “push” block is not disabled and it will execute, shifting the entire activity to the next execution window.

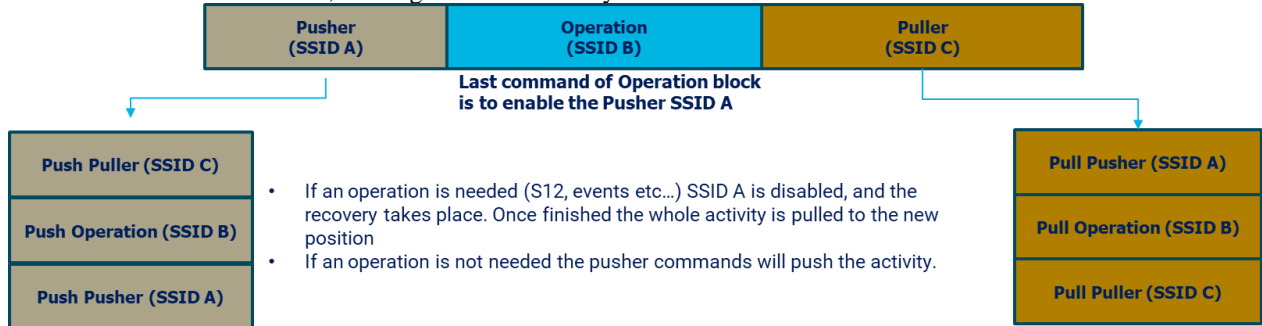


Fig. 7. Push-Pull concept

There is a limitation of this solution if only OPSO (S131) is used. If there is a request for the operation during the “pushing” activity, then the push might be interrupted and the whole activity will become desynchronised. In other words, the execution of the “pusher” and “puller” blocks is not atomic. While not ideal, one way to address this is to have the criteria disabled during those blocks.

A more efficient way to address this limitation, is to use the PMON (S12) and FMON (S142) evolutions described before, since the execution “pusher” and “puller” blocks is now atomic.

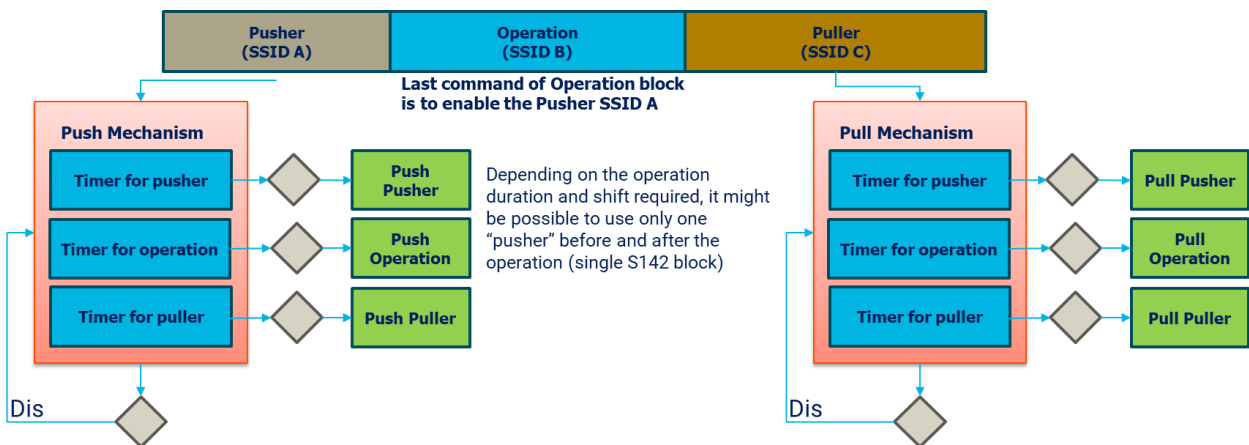


Fig. 8. Atomic push-pull concept

Another issue is the fact that due to how the OPSO (S131) is implemented on the Sentinel-3 S/C, if commands are shifted between the boundaries of the last and first orbit of the repeat cycle, the shifting will occur twice. To address this, either this limitation is accepted or the shifting in the last orbit of the repeat cycle needs to be done using different arguments for the command that performs the shifting:

- Push/pull shifting while not on last orbit of repeat cycle

- Shift between first orbit and last orbit
- Push/pull shifting not on last orbit of repeat cycle
 - Shift between first orbit+1 and last orbit

Continuing the in same thought process of having permanent operations on board, the Sentinel-3 EUMETSAT FCT decided to explore the possibility of having operations on-board that have a used defined repeat-cycle periodicity. This capability is not available in the Sentinel-3 S/C, and only available in S/C that implement PUS-C [9] , so one of ways to “emulate” this functionality is to have a mechanism that keeps the desired activity “disabled” until the target repeat-cycle is reached by making use of “gates”:

- Gate: command that if executed enables (“opens”) the SSID of the preceding gate. By default, all gates start disabled (“closed”) with the exception of the gate immediately preceding the desired activity.
- The desired activity will start disabled and is only enabled if all previous gates are opened.

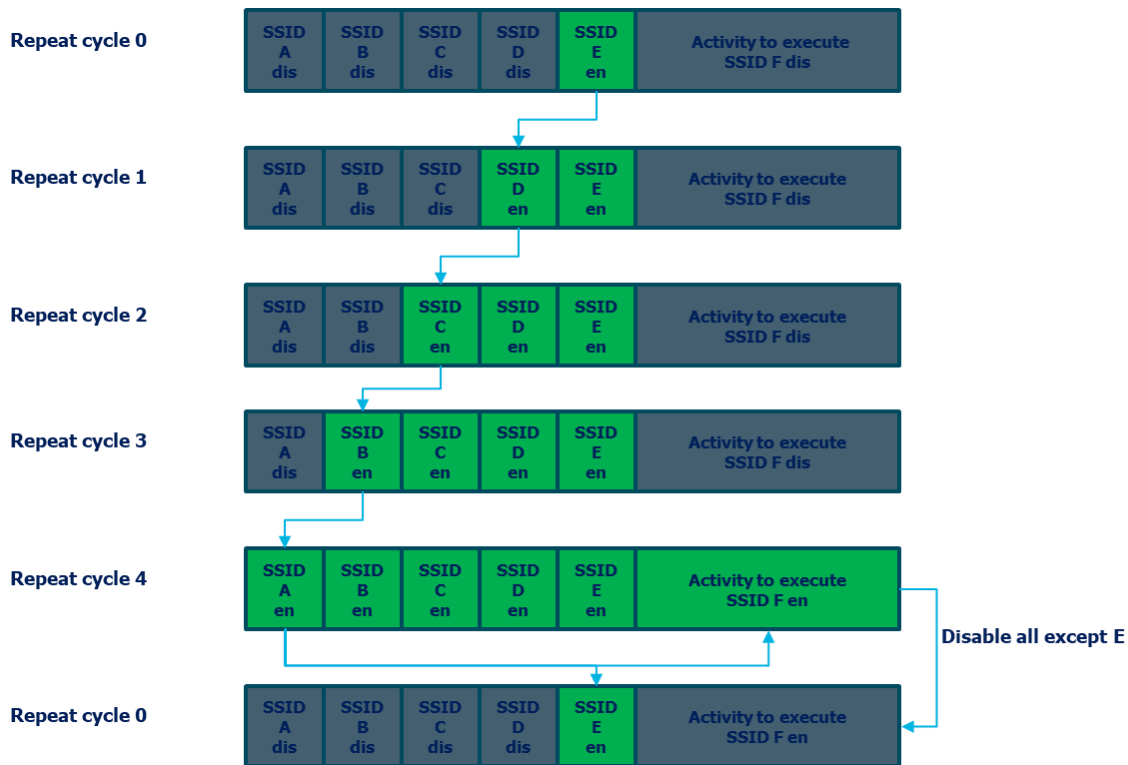


Fig. 9. Permanent intermittent operations

Mission Timeline (Service 11) evolution

Much like the evolution of the MTL (S11), it is also possible to have “dynamically” adjusted operations on-board in the MTL (S11). The difference is that the activity needs to be shifted with a “timer” and re-uplinked once executed. This can be achieved by using a combination of action timers, nested actions and step-based operations:

- A “volatile” activity is added to the mission timeline with SSID X
- A “pusher” Functional monitoring is created with an action timer in it. Once the action timer triggers, the “volatile” activity is shifted forward in time. This will cause the “pusher” functional monitoring to trigger and enable a “reset” of itself. This “reset” is achieved by having a “reset” functional monitoring that uses action timers to perform the “disable” and “enable” of the “push” functional monitoring
- If the activity is needed, then all “push” and “reset” functional monitorings are disabled and once the activity time is reached, the activity will take place
- Whenever the activity is to be uplinked to the S/C, either because it is the first time or because it executed before, the “pusher” Functional monitoring needs to be enabled afterwards.

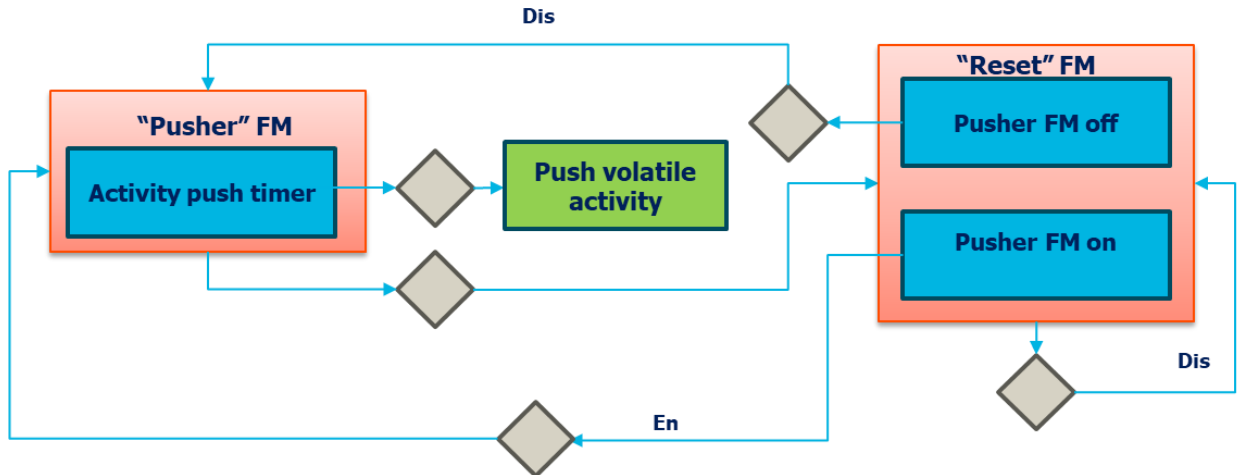


Fig. 10. Volatile automatic operations

4.2 New Automation Concept use cases

TC timeout

The Sentinel-3A and Sentinel-3B S/C do not have a TC timeout and are currently relying on commands on-board to ensure the S/C will be reachable in case of no communications are performed in a certain amount of time.

By using the above evolutions, it is possible to replace it by using action timers as well action nesting. Whenever there is a communication pass, the FCT will confirm the commanding link to the S/C is active by using an instantiated command that will activate an FMON (S142) that contains actions timers which will “restart” the running “TC timeout” timer and take care of disable the FMON (S142), thus “arming” the entire solution for the next communication pass.

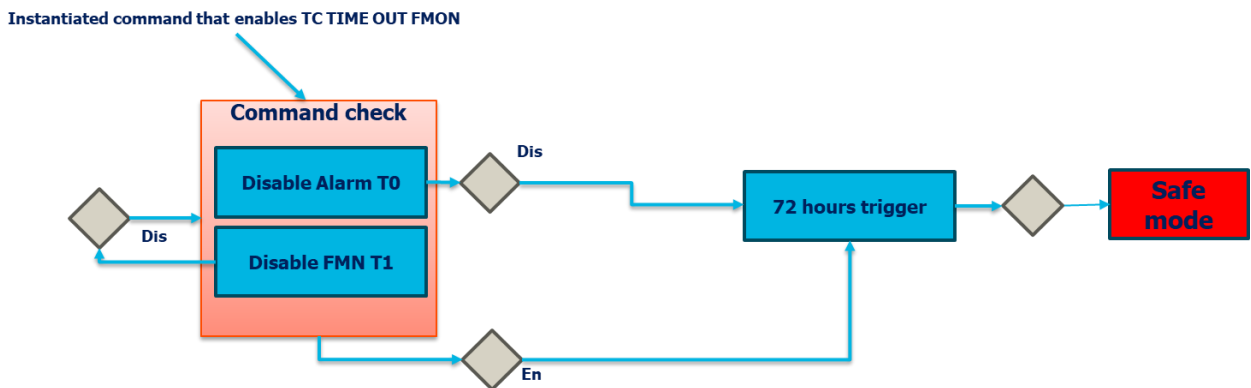


Fig. 11. TC timeout example

SLSTR CDE Power cycle

By making use of action timers, action nesting and step operations, it is possible to power cycle the SLSTR CDE [10] unit from one single “point of entry”:

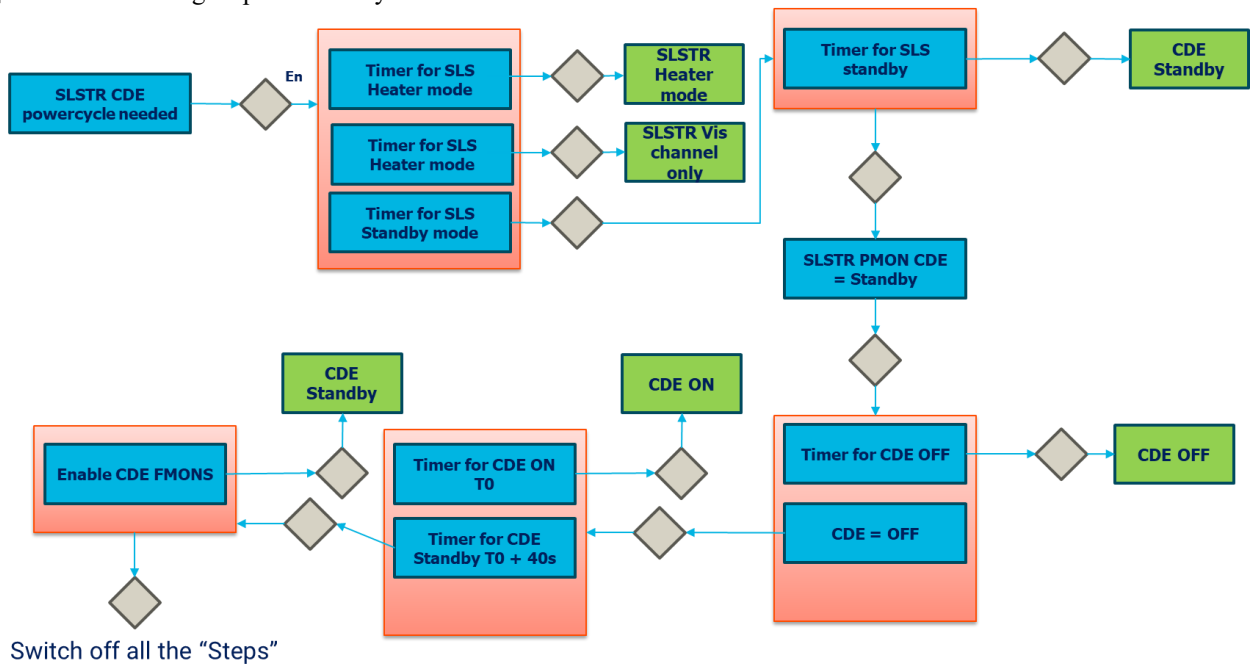


Fig. 12. SLSTR CDE automatic power-cycle

Persistent intermittent operations

The S/C in the Sentinel-3 constellation have a regular AOCS maintenance performed every 6 months. This has been done manually so far but will be replaced with a solution based on the OPSO (S131) evolution described before. This has the added benefit of being “hands-free”, with the Sentinel-3 EUMETSAT FCT only retrieving the necessary TM once the activity has taken place

Persistent dynamic automatic operations

The Sentinel-3 EUMETSAT FCT is exploring the possibility of adding autonomous recoveries on-board that require more than one action by using the OPSO (S131) evolution. This approach as the added benefit of the FCT knowing in advanced all the “recovery execution windows” and knowing in advanced how long an outage will potentially be and when the unit will be back to functioning order.

One validated recovery is the above example of the SLSTR CDE [10] power-cycle, relying on the OPSO (S131) to house the necessary commands.

Volatile dynamic automatic operations

In case the Orbital position schedule can’t be used, or for S/C that do not have it, it is also possible to have on-board that are “volatile”. The Sentinel-3 EUMETSAT FCT is planning one using this evolution for activities that need to take place before a more complex recovery can be performed. For example, if a unit is to be restarted and be left in a certain mode, this evolution could take care of that step and leave the Sentinel-3 EUMETSAT FCT to perform the rest manually, thus saving time.

The Sentinel-3 EUMETSAT FCT has validated this approach by converting the SLSTR CDE [10] power-cycle performed in the example above.

4.3 *New Automation Concept positives*

The new automation concept and the use of its building blocks allow the Sentinel-3 EUMETSAT FCT to have more control over the automation capabilities of the S/C as well as the capability to significantly expand the automation reach. With this the following positive points can be determined:

- Spacecraft safety is improved since the Sentinel-3 EUMETSAT FCT can now reliably create automatic recoveries that can have multiple actions and checks, without having to rely solely on OBCPs to do so.
- It is now possible to “emulate” action sequences, which as previously mentioned, the Sentinel-3 EUMETSAT FCT can’t create or manage
- Mission science data outage is significantly reduced, since complex recovery activities for instruments can now be added on-board
- Predictable “recovery execution widows” in time or in orbital position can now be created. These can help the Sentinel-3 EUMETSAT FCT know when potential recoveries can take place and when to expect the corresponding execution reports
- Automatic maintenance activities can now also be added on-board, either by being periodic or triggered by certain criterion
- The Sentinel-3 EUMETSAT FCT will have reduction on the workload and overall anomaly recovery stress, since many of the current recovery activities are quite demanding in terms of preparation effort and execution prowess
- By using OBCPs to augment existing and new operations, the overall effectiveness of the solutions will be improved

4.4 *New Automation concept limitations and shortcomings*

The new automation concept is not without its limitations and shortcomings.

- While the building blocks are easy to understand, linking them together can be daunting and will require specific validation effort.
- Creation of solutions using these new automation building blocks need to be done manually as there is no tool to assist the team.
- The limitations of the supporting PUS Services apply to all the solutions.
- Limited number of available slots/entries in all the building blocks
- Not persistent to safe mode
- Limited monitoring capability of internal parameters of payload units.
- Some of the above solutions would be more efficient if the Sentinel-3 S/C implement the PUS-C standard [9], since there are several updates that would be quite useful (e.g, grouping of commands in on-board command queues)

5. Future improvements of Sentinel-3 Automation Concept

The Sentinel-3 EUMETSAT FCT will explore ways to mitigate some of the OBCP limitations such that OBCPs can be used more consistently and with greater automation capability. Several OBCPs are already being developed to augment existing operations.

The Sentinel-3 EUMETSAT FCT already has several upgrades in the pipeline to the concepts presented in this paper:

- Use of “mutagenic” PMON (S12), that are dynamically modified if needed, either by ground or automatically. This is useful because of the limited number of entries in both the parameter and functional monitoring lists. Thus, having the capability to mutate, or re-use the same slot during the same “autonomous activity” is quite appealing.
- Expand on the logic gate approach by using PMON (S12) and FMON (S142) and apply it to special and/or routine ops. Some examples of possible logic gates:
 - An OR can be made by PMON (S12) grouped in to an FMON (142).
 - If any of the PMON (S12) trigger then the FMON (S142) generates the output and this can be used as result of the OR
 - An AND is possible:
 - Between expected values of two TM points can be made with a “expected value” PMON (S12).
 - Between values outside a valid range of two TM points can be made with an FMON (S142) that contains 4 PMON (S12) all with the same periodicity and filter. A “Result” or fifth PMON (S12) is to be used as the result.
 - The “Result” or fifth PMON(S12) will start disabled and with an invalid validity condition.
 - The First PMON (S12) if OOL will change the validity condition of the “Result” PMON (S12)
 - The Second PMON (S12) if OOL will enable the “Result” PMON (S12)
 - The last two PMON (S12) will be action timers that will disable the PMON and change the validity back to invalid
 - The FMON (S142) triggers last and disables itself, thus disabling all of the internal PMON (S12)

Finally, one of the future improvements to the Sentinel-3 Automation Concept is to add ground into the equation and have the ground and S/C automation work in tandem. The idea is to have a self-sufficient system that can react accordingly and make the most of the capabilities of both systems. In case there are issue that the system can’t handle, the information is escalated to the FCT for further analysis and operations.

6. Conclusion

As it has been shown, it is possible to “evolve” the use of the available capabilities of a mission like Sentinel-3 to optimise the efficiency of operations activities and minimise outages of mission data. The automation concept should not be set in stone and should be able to evolve as deemed necessary. Categories shouldn’t be enforced as these are regularly quite restrictive in their denomination and use. Instead, the mission teams should see the available automation capabilities as a toolbox that can be used to its max potential to achieve whatever the team deems necessary.

The fact that OBCPs are quite capable should not be the determining factor for the automation solution approach selection, as there are many factors that can impact this. The Sentinel-3 EUMETSAT FCT has shown that it is not necessary to rely on OBCPs to achieve a high level of automation, and that even missions that do not have OBCP capability can do so.

The PUS standard is ever evolving and bringing added benefits, but many of the new additions can be “emulated” by using the offerings of the previous PUS standard versions. One such example is the “intermittent” operations that the Sentinel-3 EUMETSAT FCT has envisioned and the PUS-C [9] implementation of the orbital position schedule.

For new missions, the recommendation is to take maximum advantage of the potential of whatever PUS standard is implemented and if possible, to implement custom automation services to meet the team's needs. It is also recommended to have a solid OBCP development environment as well as a highly capable OBCP engine on-board the S/C such that OBCPs can be used without fear of running into limitations or issues with fear or causing anomalies.

References

- [1] Donlon, C., Berruti, B., Mecklenberg, S., Nieke, J., Rebhan, H., Klein, U., Buongiorno, A., Mavrocordatos, C., Frerick, J., Seitz, B. and Goryl, P., 2012, July. "The Sentinel-3 Mission: Overview and status," In Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International (pp. 1711-1714). IEEE. 2012.
- [2] Tiago Francisco, Ed Trollope, Luca Ventimiglia, Dominique Montero, "What it has been like to fly and operate Europe's ocean and land watcher, Copernicus Sentinel 3", ", 2018 SpaceOps Conference , 28 May - 1 June 2018, Marseille, France
- [3] Dr D Milligan, P. Collins, J. Marie, E. Serpell, R. Qedar, G. Whitehead, A. Rudolph, E. Ecale, P. Tatry, "In-Flight Evolution of Onboard Automation on ESA's Gaia Mission", 2018 SpaceOps Conference , 28 May - 1 June 2018, Marseille, France
- [4] Daniel T. Lakey, Jonathan Schulster, Olivier Reboud, Michel Denis, James Godfrey, Thomas Ormston, Kees Van Der Pols, "FAST: A new MEX Operations concept, quickly!", IAA SpaceOps. Conference Proceedings 2012, Stockholm, Sweden
- [5] Daniel T. Lakey, Olivier Reboud, Alexandros Minogiannis, Johannes Bauer, Erhard Rabenau, Michel Denis, "From Drawing-Board to On-Board: A New Mission Timeline on Mars Express via OBCP", 2014 SpaceOps Conference , 5 May – 9 May 2014, Pasadena CA, United States of America
- [6] ECSS, "Space Engineering – Ground Systems and Operations. Telemetry and Telecommand Packet Utilisation Standard". ECSS-E-70-41A, 30 Jan 2003
- [7] Donati, A. "BENEFITS FROM THE PACKET UTILIZATION STANDARD IN TELEMATICS APPLICATION FOR SPACE MISSION CONTROL" *IFAC Proceedings* Volume 34, issue 9 (2001) (pp. 123-127), 2001.
- [8] Le Roy, Y., Deschaux-Beaume, M., Mavrocordatos, C., & Borde, F. "SRAL, a radar altimeter designed to measure several surface types," *Remote Sensing of the Ocean, Sea Ice, and Large Water Regions 2010*, vol. 7825, p. 782507, 2010.
- [9] ECSS, "Space Engineering – Ground Systems and Operations. Telemetry and Telecommand Packet Utilisation Standard". ECSS-E-70-41C, 15 Apr 2016
- [10] Camilletti, A., Burgess, C., Donchev, A., Watson, S., Akbar, S.W., Gamo-Albero, V., Romero-Largacha, V. & Caballero-Olmo, G., "Development of the mechanical cryocooler system for the Sea Land Surface Temperature Radiometer," *Cryogenics*, Vol. 64, pp.146-152, 2014.