ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP01

OLD AND NEW GENERATION CONTROL SYSTEMS AT ESA

M. Pecchioli, ESA/ESOC, Darmstadt, Germany

Abstract

Traditionally Mission Control Systems for spacecraft operated at the European Space Operations Centre (ESOC) of the European Space Agency (ESA) have been developed based on large re-use of a common implementation covering the majority of the required functions, which is referred to as mission control system infrastructure. The generation currently in operations has been successfully used for all categories of missions, including many commercial ones operated outside ESOC. It is however expected that its implementation is going to face obsolescence in the coming years, thus an ambitious Project is currently on-going aiming at the development and operational adoption of a new generation. The resulting infrastructure capitalizes as much as possible on the European initiative (referred to as EGS-CC, see [1]) which is progressively developing and delivering a modern and advanced platform forming the basis for any type of monitoring and control applications for space systems.

This paper is going to provide a technical overview of the various generations of the mission control infrastructure at ESOC, highlighting the main differences from technical and usability standpoints, thus describing the main lines of long-term evolution.

INTRODUCTION

Background

The operations of space assets are conducted at the European Space Operations Centre (ESOC) via the so called Mission Control System (MCS). This system therefore plays a central role in the Operational Ground Segment, which consists of the hardware and software based systems located on ground and integrated together in order to support the necessary interaction with the space segment on the one side and with other ground segments (e.g. launcher segment, payload data processing) on the other side. The main functions of the Operational Ground Segment are shown at conceptual level in the Fig. 1 below.

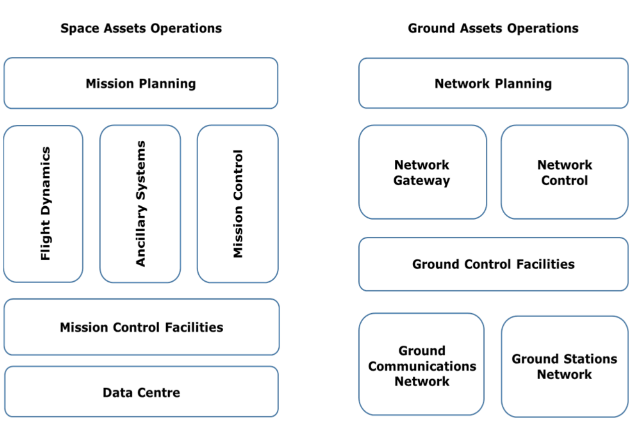

Figure 1: Operational Ground Segment (Conceptual).

It should be noted that the basic needs of the mission operators have not radically changed since the early days when spacecraft operations were first conducted. The main functions of the Mission Control systems have of course evolved but rather as a consequence of the ever increasing complexity imposed by the more and more challenging missions which need to be supported. However, the implementation of the Mission Control System infrastructure and of associated systems has been subject to much more radical evolution, primarily driven by the following factors:

- Spacecraft orbits: they dictate the frequency and the type of ground/space communication passes which can be supported. Originally only orbits enabling very long (if not continuous) visibility/contact as well as very short communication delays were supported. Nowadays the vast majority of missions rely on orbits which either provide intermittent visibility based on short passes (e.g. polar near Earth orbits) or require very long communication delays to receive and transmit the space/ground signals (e.g. deep space orbits);
- Space assets design: this is relevant to the ground systems design at several levels. The main ones affecting the mission control are: i) the protocol used to exchange telemetry (downlink) and telecommand (uplink) data; ii) the capability (generally referred to as 'monitoring and control services) supported by on-board functions which can be accessed by ground to execute mission operations; iii) volume and rates of the data exchanged with ground, in particular to download housekeeping information and payload products and iv) the level of autonomy or conversely dependence on ground operations to achieve the mission objectives;
- Operations concepts: originally mission operations strongly relied on the expertise of the ground operators and on their ability to manually execute the necessary operations. The ever increasing complexity of the mission operations as well as the necessity to minimise the associated costs during the routine phases (which heavily influence the overall costs because of the typically very long mission durations) have pushed for more modern approaches relying on a higher level of interaction (e.g. executing procedures rather than sending individual commands), on a higher level of automation (operations executed by ground applications rather than human operators) as well as a on higher level of on-board autonomy (e.g. time or event driven operations autonomously executed on-board);
- Software technologies: quite obviously the development of Mission Control Systems relies on off-the-

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP01

shelf software technologies and thus its design and implementation is directly impacted by the technological evolutions in this area;
- IT infrastructure: again, the Mission Control Systems make use of off-the-shelf infrastructure systems providing computer and network resources and thus its design is heavily affected by the underlying technology which is going to be used.

Mission Control Systems Infrastructure Design

Due to the tight link between the design of the operational systems located on ground and the one of the space segments, as well as because of the direct dependency on the continuously evolving software and computing technologies, in the majority of space organisations Mission Control Systems have been and still are developed on the basis of a completely mission specific lifecycle (specification, design, development, validation and operations). On the contrary, the complete Operational Ground Segment of ESA missions has been traditionally designed and developed by relying on domain (but not mission) specific infrastructure i.e. implementations which can be configured, customised and adapted in order to serve the full palette of missions to be operated. This approach is not so much driven by technical considerations but rather by strategic and programmatic considerations, such as:

- Ensure stable, predictable and low system development and maintenance costs. This is best achieved by sharing a large fraction of these costs across multiple missions;
- Facilitate cost reductions which can be obtained by reducing the preparation time of new missions. Having a ‘nearly ready’ solution for the critical ground systems ensures minimum design and development time;
- Facilitate risk reductions when implementing the ground segment for new missions. The provision of a proven common infrastructure which can be tailored for the needs of a specific mission allows a significant mitigation of this risk;
- Minimise familiarisation effort of system developers and mission operators. In large space organisations the highly specialised staff are typically ‘re-used’ across missions, thus this objective is achieved by minimising the differences between different missions.

It is fair to state that the potential benefits illustrated above can only be really exploited if the Mission Control System is carefully and adequately designed. This leads to the need of considering important design drivers which would play a less prominent role when dealing with mission specific systems, namely:

- Support of heterogeneous space systems: the ground/space interaction at data level takes place by means of Telemetry (downlink) and Telecommand (uplink) data transmission protocols. The progressive adoption of standardised solutions has reduced the amount of different protocols which need to be supported. However, mission operations and therefore the design of Mission Control Systems are also affected by the type of interaction which is available to enable appropriate access to the on-board functions. This is an area where European satellites have also achieved a good level of standardisation (through the so called Packet Utilisation Standard), thus enabling a wide adoption of common solutions on ground;
- Support of multiple mission types: this is essential in order to enable the support of a wide variety and thus a large number of missions, without restrictions related to the mission type. Typically, the operations concept and thus the ‘type’ of unmanned missions is mostly affected by the spacecraft orbit, in particular distinguishing between long passes/short delay missions (e.g. geostationary satellites), short passes/short delay missions (e.g. Earth observation satellites in polar orbits), long passes/long delay missions (e.g. deep space). The support of the various mission types may imply significant variations in the functions to be supported on ground in order to execute mission operations;
- Standardised interfaces: the control system plays a central role in the execution of mission operations and as such needs to support a significant number of file based and programmatic interfaces with the other ground systems involved in the operations preparation, planning, execution and evaluation. In order to avoid the need of mission specific adaptations and revalidation of interfaces serving the same purpose, the design needs to accommodate for a generic approach which is re-usable for all missions and can be preserved in spite of the unavoidable technological and functional evolution of the ground systems. In addition, interfaces between facilities which are not necessarily managed by the same organisation (e.g. control centres and ground stations) need to ensure inter-operability and thus are subject to international standards managed by organisations such as the CCSDS;
- Modular architecture: in order to enable the parallel development of generic and mission specific parts as well as the effective maintenance and re-use of a generic infrastructure it is essential to decompose it in a modular and hierarchical manner achieving a proper separation of concerns of the various constituting elements;
- Generic and extensible functionality: genericity applies to all functions whose implementation can be shared by all target applications, extensibility is instead the mechanism enabling the integration of application specific features without source code level modifications of the infrastructure implementation;
- Capability to tailor: the control functions are designed using a ‘data driven’ approach in order to enable their tailoring for a given controlled system without code modifications. The characteristics of the monitoring and control data exchanged between ground and space (generally referred to as Telemetry and Telecommand

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP01

data) are defined in a 'database' enabling the conversion from human readable artefacts into data units which are processed and eventually encoded for transmission in the uplink (control) and downlink (monitoring) directions;

- Functional richness: in order to become an attractive solution for developing mission control systems, a generic infrastructure needs to provide a rich set of functions which go beyond the specified user needs and support the safe and 'comfortable' execution of mission operations;
- Layered design: efficient maintainability of the various contributing elements can only be effectively achieved if a clean separation between the re-usable/generic parts and the mission specific extensions/adaptations is introduced and strictly respected throughout the utilisation phase. A strictly layered design is functional to this objective, based on a separation between interfaces and services accessible to the higher layers and internal implementation. The ideal design is such that no modification of the infrastructure implementation is required in order to develop and deploy a fully-fledged mission control system;
- Operations abstraction: in order to simplify the execution of mission operations it is important that an appropriate level of abstraction is supported and adopted. This allows the operators to concentrate on the 'what' needs to be done and to remove the focus from the low level implementation details;
- Support of automation: one of the cost drivers for long-lasting missions is the support of routine operations. Support automation at all levels (from the system artefacts deployment up to execution of planned operations) is considered an essential feature in order to enable proper control of these costs;
- High performance: the various utilisation scenarios to be served by a generic infrastructure often introduce orthogonal performance drivers (e.g. data storage and distribution rates, data volumes, data processing rates, number of parallel user applications). It is also important not to introduce dependencies on expensive hardware resources which may not be compatible with low-cost missions;
- Longevity and long term maintainability: space programs are characterised by a very long lifetime. In addition, a generic infrastructure is only really effective if it is adopted and used by many missions. These aspects introduce the need of maintaining and sustaining the generic infrastructure for various decades and without leading to explosive costs. This aspect has important implications not only at technical level e.g. it is of paramount importance to ensure that relevant expertise is continuously created and sustained;
- Portability/technology isolation: the implementation of a control system infrastructure heavily relies on re-used off-the-shelf technologies. As a consequence of its required longevity, it is essential that the implementation of the business logic is isolated as far as possible from the re-used technologies. This simplifies the portability to newer versions of continuously evolving products (which rarely ensure full backwards compatibility in the long term);
- Scalability: mission requirements may significantly differ in terms of resources required by their control system. An example is the support of large constellation of many spacecraft compared with the support of small 'low cost' satellites. In order to be able to re-use the same generic implementation it is therefore of paramount importance that this can make an effective use of scalable computing and storage resources but at the same time without introducing a dependency on highly expensive hardware.

The features listed above have driven the design of the various generations of mission control system infrastructure developed at infrastructure. Nonetheless significant differences in the obtained design can be observed. An analysis of these differences and their rational is the subject of the next sections.

INFRASTRUCTURE EVOLUTION

In this section we briefly introduce the main Mission Control System infrastructure generations which have been developed at ESOC and attempt a high-level comparison between their main design characteristics.

During the more than 40 years during which this 'infrastructure based' approach has been adopted, several distinct generations of Mission Control System infrastructure have been designed and developed, namely:

- Multi-Satellite Support System (MSSS), that was operationally used since the '70s up to the end of the '90s;
- Spacecraft Control Operations System (SCOS) 1, that was operationally used since the beginning of the '90s and is still in use in some legacy missions;
- SCOS II (later renamed into SCOS-2000), that was operationally used since the beginning of this century and still forms the basis for the vast majority of control systems being used at ESOC for currently operational missions and the ones to be launched next missions;
- Mission Control Core based on Common Core (M4C), that provides equivalent features as SCOS-2000 and its complementary systems, however based on the recently developed control framework EGS-CC (European Ground System – Common Core).

The following sections analyse the evolution of the control system infrastructure by highlighting the main design aspects of its various generations.

System Context

The role of the Mission Control System (MCS) in the Operational Ground Segment has not significantly changed throughout the years. Since the very beginning the MCS

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP01

plays the role of the only ground system which is responsible for managing all interactions with the space assets under control (through the ground stations network). However, the MCS interactions with other ground systems has significantly evolved throughout the years. A progressive adoption of open architectures and the support of off-line (file based) as well as on-line (API based) external interfaces has responded to the continuously increasing needs of interactions with other systems during the various operations phases, namely:

- Preparation: this covers the production and verification of all user specified operational artefacts enabling the execution of mission operations (e.g. spacecraft TM/TC characteristics, procedures, displays). The user applications supporting the mission operations preparation have significantly evolved, starting from very basic text editors, going through a proliferation of heterogeneous tools based on stand-alone Windows applications providing advanced visualisation capabilities as well as a higher level of usability and eventually implemented on the top of an expandable Eclipse-based framework (the Operations Preparation Environment, OPEN) offering the ability to rapidly develop and integrate specific editors for the various data types to be generated and to use them in a distributed (but centrally managed) environment;
- Planning and Scheduling: this covers the processing of the mission exploitation requests as well as the space segment housekeeping needs in order to produce an optimised sequence of operations to be executed in order to maximise mission exploitation and duration. In the early days this function was 'manually' supported by mission engineers, with the limited support of ad-hoc created tools. Nowadays a completely generic infrastructure has been developed, managing the reception and processing of requests from the Mission Exploitation Centres (e.g. Scientific Operation Centres), the constraints imposed by the limited on-board resources and ground/space communication (e.g. visibility passes) and the needs of executing flight dynamics (such as orbit manoeuvres) and spacecraft housekeeping (in order to produce a detailed sequence of commands to be uploaded to the ground and spacecraft schedules for time-tagged execution. The MCS infrastructure currently supports the reception of these commanding artefacts and provides feedback about the upload and execution status;
- Critical operations: this covers the execution of mission operations during critical phases (such as the LEOP, Launch and Early Orbit Phase), which takes place under the direct supervision of the mission operators. Conceptually speaking very limited changes have been observed in this area. The operators keep track of and acts upon the state of the spacecraft through various types of displays, such as: i) snapshot displays (e.g. alphanumeric displays with user specified layout, summary of non-nominal states, user-defined synoptic displays) showing the space segment state in live or at a specified time in the past; ii) log displays, showing the history and the latest state of data units of operational interest through scrolling tables showing the relevant attributes of each instance of a given data type (e.g. commands, reports, events, messages, alerts, control actions, state transitions, file transactions); iii) trend displays (e.g. plot displays), providing the capability to observe and analyse the evolution over time of state parameters; operations stack displays, enabling the user to create sequence of operations to be executed and manage their execution under manual supervision or through a semi-automated approach; object details displays, providing complete visibility of the definition and state of a given object (e.g. a parameter, a command, an event, a report) in an organised layout supporting editing capabilities where relevant. The type of displays depicted above have been devised since the very beginning with their complementary function/role. What has radically changed is the amount of displays which the various users can access at a given time (MSSS only supported the ability to open full-screen displays, up to three!) and the level of sophistication of the user interactions with the displays. This aspect is closely bound to the capabilities available in off-the-shelf technologies, the latest generations being based on powerful frameworks supporting the development and deployment of smart clients, such as the Eclipse Rich Platform;
- Routine operations: this covers the execution of regularly planned operations originating from the mission utilisation and space segment housekeeping needs. The early generations of the Mission Control Systems were strictly designed to support the 'manual' (operator driven) and real-time execution of control operations. The design of control features has progressively moved towards the support of autonomous space segments (e.g. time-driven, event-driven, procedure-driven and goal-driven release of commands from on-board applications) as well as ground automation (e.g. release time-tagged schedules of ground activities, event-action relationships supporting automated reactions, automatic procedure executions for repetitive operations, fully automated execution of operations schedule generated by the mission planning, automatic detection and notification to remote operators of alert conditions). The very first generation of mission control infrastructure (MSSS) only supported the capability to authorise in advance the release of multiple consecutive commands separated by specified time intervals. The latest mission control infrastructure generation (M4C) natively supports the capability to execute any type of control action (generally referred to as Activities) through ground or on-board triggers based on delta or absolute times, events or procedures. Another area where the execution of routine operations has fundamentally evolved relates to the adoption of the so called File based Operations (see [2]) concept. Traditionally, spacecraft operations have been executed relying on very low level protocol data units exchanged

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP01

between ground and space (e.g. packets). The development and adoption of international standards specifying protocols supporting the delivery of files in both directions (upload, from ground to space and download, from space to ground) has opened the door to interaction concepts based on a much higher level of granularity (e.g. complete schedule increments uploaded and activated with a minimum set of operations). The latest generations of mission control infrastructure (SCOS-2000 and M4C) both support file based operations concepts, thus providing the mission operators with an efficient environment to execute routine operations;

- Evaluation: this covers all ground tasks taking place after operations execution to manage, disseminate and analyse/report the relevant data. This area has observed a significant evolution, to accommodate the needs of more and more distributed teams by exploiting the more and more powerful software technologies in this domain, including the recently developed Big Data capabilities.

System Design

The system level architecture of the mission control infrastructure has significantly evolved throughout the various generations.

- Modularity: as described in the previous section, the modularity of the mission control infrastructure is an essential design aspect as it eases maintenance, enables parallel/distributed development of the different modules as well as a clean separation between generic and mission specific contributions. The infrastructure evolved from a completely monolithic implementation up to the truly component based system organisation. In M4C each system component consists of self-standing artefacts, such as design model, 'external' interfaces, test environment and procedures, configuration data items, tailoring data enabling user driven as well as automated monitoring and control operations;
- Deployment: in this area the mission control infrastructure design has adopted significantly different solutions, following the relevant trends of IT technologies. MSSS was based on a completely centralised architecture, whereby two redundant instances of the system running on main-frame infrastructure supported all missions and provided remote access via hardwired workstations. SCOS-2000 is based on a fully distributed client-server architecture, initially based on 'very fat' clients, providing local support of user dedicated functions, then progressively re-engineered in order to enable remote access via dumb terminals as well as to introduce a clearer separation between the user interaction layer and the underlying business logic of user dedicated applications. M4C is designed on the basis of a 'conceptually centralised' deployment concept, whereby only the user interaction layer is supported by user dedicated applications remotely accessible via 'installation-free' thin clients. The centralised functions are deployed over distinct tiers, each supporting consolidated and scalable solutions (e.g. software defined networks, cloud computing, cluster based storages). It is important to note that this architecture, in combination with the strictly layered system organisation (e.g. management of system sessions), also opens the door to more sophisticated deployments, whereby distinct missions share a large part of the mission control functions with other missions ('Multi-mission' deployments) in order to minimise the effort associated to the preparation, integration and validation processes;
- Scalability: this is directly affected by the deployment options described above. MSSS wasn't really scalable, in that the deployment of mission control systems took place on the same 'resource-bound' computer platform for all missions. SCOS-2000 based Mission Control Systems are deployed on dedicated physical machines and thus each of them is bound by the limits of the underlying supporting hardware platform. M4C will offer the widest range of deployment solutions, ranging from completely dedicated ones (e.g. physical machines equipped with local storage) up to completely shared ones (e.g. Multi-mission virtual data centres), thus enabling optimised access to the available computing and data storage resources and exploit their scalability;
- Tailoring: a key aspect of a generic mission control infrastructure is its capability to be adapted and extended for a given specific need such as a particular mission. All generations are characterised by their high level of tailorability to provide the user with the capability to specify the exact structure of the monitoring (Telemetry) and control (Telecommand) data units which are supported by a specific spacecraft. Little conceptual evolution has taken place in this area, although the level of sophistication and complexity of the spacecraft data structures and the associated ground data definitions has progressively and significantly increased over time;
- Adaptability/Extensibility: this covers the ability to adapt a common implementation to a specific environment, to modify the behaviour of the generic functions to meet the specific aspects of a given mission and to extend the commonly provided functionality to add mission specific features. Clearly the early generation of the systems were heavily relying on the adoption of common solutions for all missions (e.g. standardised TM/TC interface protocols), whereby the system configurability would be sufficient to ensure compatibility with the specific aspects of a given mission and its control environment. More and more the mission control infrastructure has then evolved towards the capability to be customised and extended at software level. A key design difference in this area relates to the ability to introduce mission specific functions or even behavioural modifications of the common functions

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP01

without source code level modifications of the infrastructure implementation. Enabling and enforcing this clear separation between generic and specific implementations has been the design focus of M4C. This design feature will significantly contribute to the efficiency of the integration, validation and maintenance processes of Mission Control System sharing the same common implementation. Another area where significant improvements have been introduced in the recent generations of mission control infrastructure is the support of 'user defined customisations and functionality extensions'. More and more the mission operators request the capability to expand the user accessible functions by autonomously developing e.g. operational scripts, data processing routines, control displays, advanced monitoring displays, etc.;

- Inter-operability: as stated in previous sections, the Mission Control Systems play a central role in the ground segment and act as the only monitoring and control counterpart of the space segment. This implies the support of a significant number of complex and critical interfaces between expensive systems, not necessarily governed by the same organisation even within the boundaries of an individual mission. For this reason, traditionally the ground/space interfaces have been the subject of international standards aiming at ensuring inter-operability of ground stations and space assets. Progressively, more and more standardised protocols at data level have been developed and adopted, thus also enabling the seamless adoption of the same control system implementations across different missions. In particular, the TM/TC data exchange protocols between control centres and ground stations have been standardised by the CCSDS through the so called Space Link Extension (SLE, see [3]), enabling seamless access to space assets through the international network of ground stations supporting these protocols. Still lagging behind is the adoption of standardised solutions between ground systems involved in the execution of mission operations (e.g. mission planning, flight dynamics, payload data processing). The various mission control infrastructure generations have systematically adopted all the standardised solutions supported by the interfacing systems, thus supporting a continuously increasing level of interoperability across ground and space assets. In particular, both SCOS-2000 and M4C provide native support to 'service level' operations of spacecraft based on the ECSS Packet Utilisation Standard (PUS, see [4]), which consists of a rich set of on-board functions providing the mission control function on ground with advanced monitoring and control capabilities;
- Performance: significant increases in the performance demand and support has driven the evolution of the mission control infrastructure. The MSSS could support TM data reception and processing in the order of few Kilobits per second, whereby the Gaia Mission Control System based on SCOS-2000 can sustain an incoming data rate of more than 10 Megabits per second and a full TM data processing rate at about 1 Megabits per second. This represents an increase of several orders of magnitude, which also gives an indication of the corresponding increase in computing resources which need to be made available and exploited. Also the amount of parallel users requiring simultaneous access to the system (e.g. during LEOP) has faced an important increase. MSSS could support up to 20 user workstations in the full centre (!) for all missions supported in operations, whereby during LEOPs of today's missions it is quite common that more than 50 users are connected to a single Mission Control System;
- Technology: the implementation of the mission control infrastructure relies on large re-use of suitable 3rd party technologies supporting the development and operation of critical and large applications. Considering the rapid evolution of these technologies in the last decades, it is not difficult to imagine that the various generations of mission control infrastructure have followed the main trends and thus have been based on completely different technology stacks and associated architectures. This applies in particular to: i) programming, scripting and modelling languages (e.g. Fortran, C, C++, Java, Groovy, XML, XMI); ii) middleware and run-time frameworks (e.g. broker based architectures, components framework, plug-in architectures, message buses); iii) user interfaces (fat clients, rich clients, smart clients, thin clients); iv) data storage engines (binary files, relational databases, no-SQL, Big Data); v) operating systems (Unix, Linux, Windows). In particular, the latest generation M4C is based on a modern stack of technologies which ensures high level of platform independence and portability, support of multi-tier deployments and scalable exploitation of the available computing and storage resources;
- Maintainability: this relates to all design aspects enabling an efficient maintenance process of the generic infrastructure implementation itself as well as the one of Mission Control Systems based on it. In M4C significant effort has been devoted to achieve good level of long-term maintainability by: i) adopting 'portable' technologies and minimising the direct dependencies on off-the-shelf technologies (to minimise the effort to adapt to their continuous evolution); ii) adopting widely used and mature languages and associated technologies, for which a large pool of experts is available on the labour market; iii) enforcing a modular design whereby each module is only 'visible' to other modules through well-defined 'service interfaces', thus removing any cross-dependency on internal implementation; iv) providing extension and customisation hooks, to avoid the need of source code level integration and enable a strictly layered maintenance approach; v) introducing a high-level of automation in the off-line processes related to the production, deployment and, most important, validation of the operational systems. The last three aspects in particular

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEAPP01

represent a significant improvement of the last generation mission control infrastructure M4C in comparison with the previous generation SCOS-2000.

CONCLUSION

Since the early days of spacecraft operations support, Mission Control Systems at ESOC are developed on the basis of a common implementation shared by multiple target missions, referred to as 'mission control infrastructure'. Various generations of the mission control infrastructure have been developed so far, which significantly differ from each other in many aspects, although providing similar functions. This paper has introduced the most important design features of the mission control system infrastructure in general and analysed the main evolution of its implementation. It can be concluded that, in spite of its relative 'slowness', the long-term evolution of the ESA mission control infrastructure has been able to accommodate the ever increasing missions' demand in terms of performance, functional richness, usability and cost efficiency. This has been possible through a 'continuous' (within generations) as well as 'step-wise' (across generations) process of enhancing the system capabilities, by capitalising on progressively gained operational and engineering experience as well as on the more and more powerful software technologies which have become available off-the-shelf.

ACRONYMS LIST

CCSDS	Consultative Committee for Space Data Systems
ECSS	European Co-operation for Space Standardisation
EGS-CC	European Ground Systems Common Core
ESA	European Space Agency
ESOC	European Space Operation Centre
IT	Information Technology
LEOP	Launch and Early Orbit Phase
M4C	Mission Control Core based on Common Core
MCS	Mission Control System
MSSS	Multi-Satellite Support System
OPEN	Operations Preparation ENvironment
PUS	Packet Utilization Standard
S2K	SCOS-2000
SCOS	Spacecraft Control Operations System
SLE	Space Link Extension
SQL	Structured Query Language
TM	Telemetry
TC	Telecommand
XMI	XML Metadata Interchange
XML	eXtensible Mark-up Language

REFERENCES

[1] M. Pecchioli, J.M. Carranza, "Highlights of the European Ground Systems - Common Core Initiative", in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, Barcelona, Spain, October 2017, paper THCPL02, pp. 1175-1181. doi:10.18429/JACoW-ICALEPCS2017-THCPL02

[2] M. Pecchioli *et al.* "Towards File based Operations", in *SpaceOps 2010 Conference*, Huntsville, Alabama, USA, April 2010. doi.org/10.2514/6.2010-2190

[3] Space Link Extension, Consultative Committee for Space Data Standardisation, https://public.ccsds.org/Publications/default.aspx

[4] ECSS-E-70-41A Packet Utilisation Standard, European Co-operation for Space Standardisation, http://ecss.nl/standards/