Chapter 7

CCSDS Standardization in Spacecraft Monitoring and Control

Mario Merri*
European Space Agency, Darmstadt, Germany

I. Introduction

THIS chapter presents a set of concepts, reference architecture, and service framework for spacecraft monitoring and control. It has been prepared by the Spacecraft Monitoring and Control (SMC) Working Group (WG) of the Consultative Committee for Space Data Systems (CCSDS) Mission Operations and Information Management Systems (MOIMS) area.

The SMC WG is a reasonably young initiative of the CCSDS, which was initiated in December 2003. It started as CCSDS Bird of a Feather (BOF) to investigate Q1 whether the space community had enough interest in the topic. Since then, the demonstration of interest and the participation have been significant, which justified the establishment of a dedicated CCSDS Working Group, the SMC WG. As of today, the WG consists of the following 10 space agencies, which are actively contributing: ASI (Italian Space Agency), BNSC (British Space Agency), CNES (French Space Agency), CSA (Canadian Space Agency), DLR (German Space Agency), ESA (European Space Agency), FSA (Russian Space Agency), INPE (Brazilian Space Agency), JAXA (Japanese Space Agency), and NASA. Q2

II. Goals

A. Problem

There is a general trend toward increasing mission complexity at the same time as increasing pressure to reduce the cost of mission operations, both in terms of initial deployment and recurrent expenditure. Often, closed or "monolithic" mission operations system architectures are used, which typically are not component-based in nature and do not offer open interfaces. They do not allow the redistribution of functionality between space and ground, or between nodes of the ground system. This lack of architectural openness leads to 1) lack of interoperability between agencies, 2) lack of reuse between missions, 3) increased cost of mission-specific development and deployment, 4) unavailability of commercial generic tools, 5) inability to replace implementation technology without major system redesign, and 6) lack of operational commonality between mission systems – increased training costs. The result is many parallel system infrastructures that are specific to a given family of spacecraft or operating agency, with little prospect of cross-fertilization between them.

B. Service Framework Approach

Service-oriented architecture (SOA) is an approach to system design that relies not on the specification of a monolithic integrated system, but instead on the identification of smaller, modular components that communicate only through open, published, service interfaces. ~~This~~ framework of standard services enables many similar systems to be assembled from compliant "plug-in" components. These components may be located anywhere, provided they are connected via a common infrastructure. This allows components to be reused in different mission-specific deployments: between agencies, between missions, and between systems.

If services are specified directly in terms of a specific infrastructure implementation, then they are tied to that technology. Layering of the services themselves allows the service specifications to be made independent of the underlying technology. Specific technology adapters enable the deployment of the service

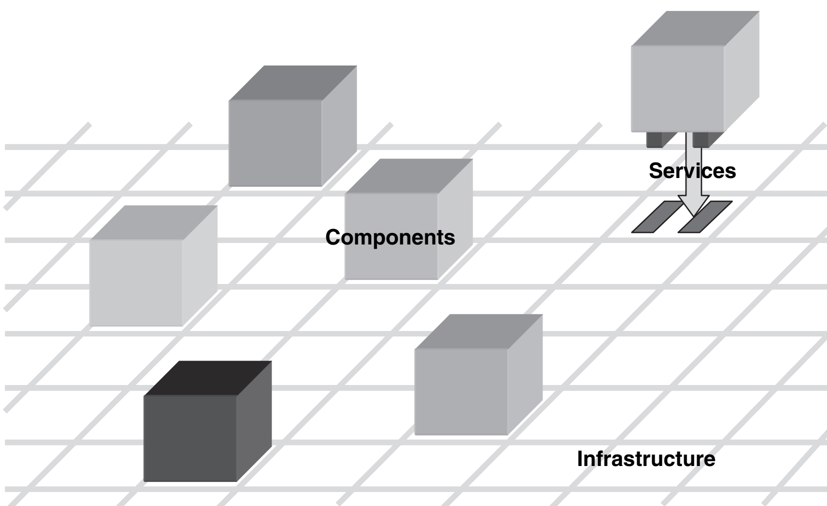

Fig. 1 Service-oriented architecture. Plug-in components communicate only via standard service interfaces through a common infrastructure. The service framework is itself layered and independent of the underlying infrastructure implementation. Q4

framework over that technology. This in turn makes it possible to replace the infrastructure implementation as well as component implementations. It is also possible to transparently bridge between different infrastructure implementations, where these are appropriate to different communications environments (e.g., space or ground) or simply reflect different agencies' deployment choices.

The approach is intended to be evolutionary and not revolutionary, and the service framework must also respect legacy systems. Where an integrated legacy system performs the function of several service framework components, its internal architecture and implementation does not have to be changed. Only those interfaces it exposes to other systems need be "wrapped" to make them compliant with the corresponding service interfaces. The service framework offers a range of interoperable interfaces, from which the most appropriate can be selected: compliance is not dependent on supporting them all. In this way legacy systems can be reused in conjunction with other compliant components to build a mission-specific system.

It is also important to note that the approach does not prescribe the components themselves or their implementation. Only the service interfaces between components are standardized. This allows for innovation, specialization, and differentiation in components, while ensuring they can be rapidly integrated into a system. However, for the service framework to be effective, it must ensure that semantically meaningful information associated with mission operations can be exchanged across the service interfaces, not merely data.

C. Potential Benefits

Standardization of a mission operations service framework offers a number of potential benefits for the development, deployment, and maintenance of mission operations infrastructure:

1) Increased interoperability between agencies, at the level of spacecraft, payloads, or ground segment infrastructure components.

2) Standardization of infrastructure interfaces, even within agencies, leading to reuse between missions and the ability to establish common multimission infrastructure.

3) Standardization of operational interfaces for spacecraft from different manufacturers.

4) Reduced cost of mission-specific deployment through the integration of reusable components.

5) Ability to select the best product for a given task from a range of compatible components.

6) Greater flexibility in deployment boundaries: functions can be migrated more easily between ground segment sites or even from ground to space.

7) Standardization of a limited number of services rather than a large number of specific intercomponent interfaces.

8) Increased competition in the provision of commercial tools, leading to cost reduction and vendor independence.

9) Improved long-term maintainability, through system evolution over the mission lifetime through both component and infrastructure replacement.

III. Scope

A. Mission Operations

The term mission operations is used to refer to the collection of activities required to operate spacecraft and their payloads. It includes 1) monitoring and control of the spacecraft subsystems and payloads, 2) spacecraft and ground segment performance analysis and reporting, 3) planning, scheduling, and execution of mission operations, 4) orbit and attitude determination, prediction, and maneuver preparation, 5) management of onboard software (load and dump), and 6) delivery of mission data products.

These are typically regarded as the functions of the mission control center (MCC) and are performed by the mission operations team, supported by the mission control system (MCS). Activities concerned with the exploitation of mission data, including its archiving, processing, and distribution, are considered outside the scope of mission operations. Increasingly, mission operations functions may be distributed between collaborating agencies and ground segment sites, or partially delegated to autonomous functions onboard the spacecraft itself.

The mission operations service framework is concerned with end-to-end interaction between mission operations application software, wherever it may reside within the space system. It is specifically not concerned with the provision of services for data transport or persistence (storage). It is, however, a user of such services.

B. System Boundaries and Interoperability

The needs of individual missions will require flexible collaboration between agencies. Although operational responsibility for a satellite normally resides with its owner agency, it may carry payloads, probes, or landers that are owned and operated by third parties. It is also the case that satellites from several different manufacturers may be owned and operated by a single agency. The demands for greater onboard autonomy and increasing onboard processing power will also allow migration of functionality onboard the spacecraft. This exposes more complex mission operations interactions to the space-ground interface. Standardization will enable the development of re-usable infrastructure in both ground and space segments.

Where an interface is exposed between agencies, it becomes an interoperable interface and a candidate for standardization. The variability of mission operations system configuration outlined previously means that most of the main interfunctional interfaces of mission operations could be either internal or external to a given system. Even within an agency or other operating organization, there are benefits to the standardization of mission operations services, as outlined in Sec. II.C. The concept for a mission operations service framework allows for incremental standardization as follows:

1) Priority is given to services that are currently exposed at interoperability boundaries.

2) Services exposed at key internal interfaces within the infrastructure of multiple agencies will be standardized to encourage the development of reusable infrastructure components.

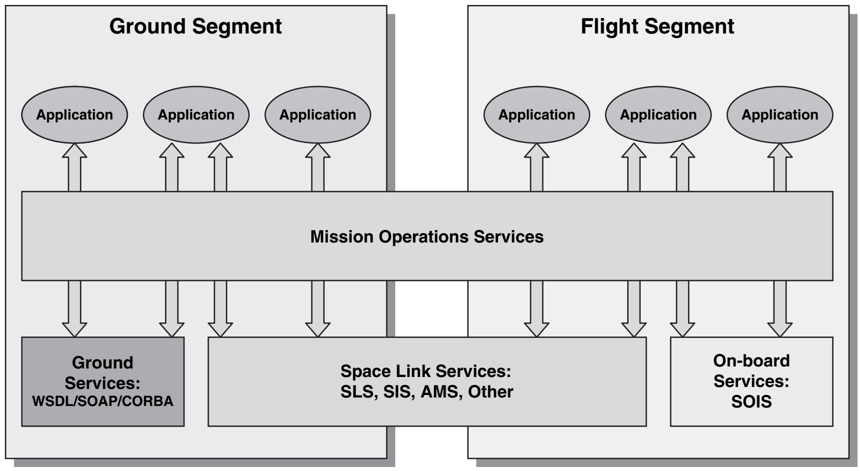

Fig. 2 Relationship of mission operations services to other CCSDS standards.

3) Finally, an initial set of services are identified to allow future evolution of interoperable interfaces with increased complexity of missions and onboard autonomy. This set is by no means predefined, and the working group anticipates that it will change as the work progresses.

C. Relationships to Other Standards

The mission operations service framework addresses end-to-end interaction between applications that reside within both the space and ground segments. The underlying transport services over which mission operations services are carried may be different depending on the nature of the communications path:

1) Between space and ground. This is expected to use CCSDS Space Link Services (SLS), packet telemetry and telecommand (TM/TC), and optionally CCSDS Space Internetworking Services (SIS). In particular, the proposed Asynchronous Messaging Service (AMS) offers a messaging layer over which the protocol messages of the mission operations service framework could be carried. Similarly, the CCSDS File Delivery Protocol (CFDP) may be used to support file transfer. Q5

2) Within the ground segment. Wider industry standard middleware services may be used, such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Java Remote Method Invocation (Java-RMI), Java Message Service (JMS), or Common Object Request Broker Architecture (CORBA). Alternatively, AMS could be used over TCP/IP. Similarly, the file transfer protocol (FTP) may be used to support file transfer. Q6

3) Onboard the spacecraft itself. CCSDS Spacecraft Onboard Interface Services (SOIS) could be used.

Some applications may bridge between one underlying communications environment and another, e.g., a ground mission control system may use packet

TM/TC to communicate with the spacecraft, but SOAP to forward parameter status to a payload control center. CCSDS Cross Support Services as the Space Link Extension (SLE) ones, which are not shown in the diagram, may also be used transparently to the mission operations services to extend the space link from ground stations to a mission operations center.

IV. Summary of Approach

The CCSDS Spacecraft Monitoring and Control (SMC) Working Group has developed a concept for a mission operations (MO) service framework, which follows the principles of service-oriented architectures. It defines an extensible set of end-to-end services that support interactions between distributable mission operations functions—software applications specific to the mission operations domain.

A. Context of SMC MO Service Framework

The MO service framework sits between application software specific to the domain of spacecraft mission operations and the underlying technology used

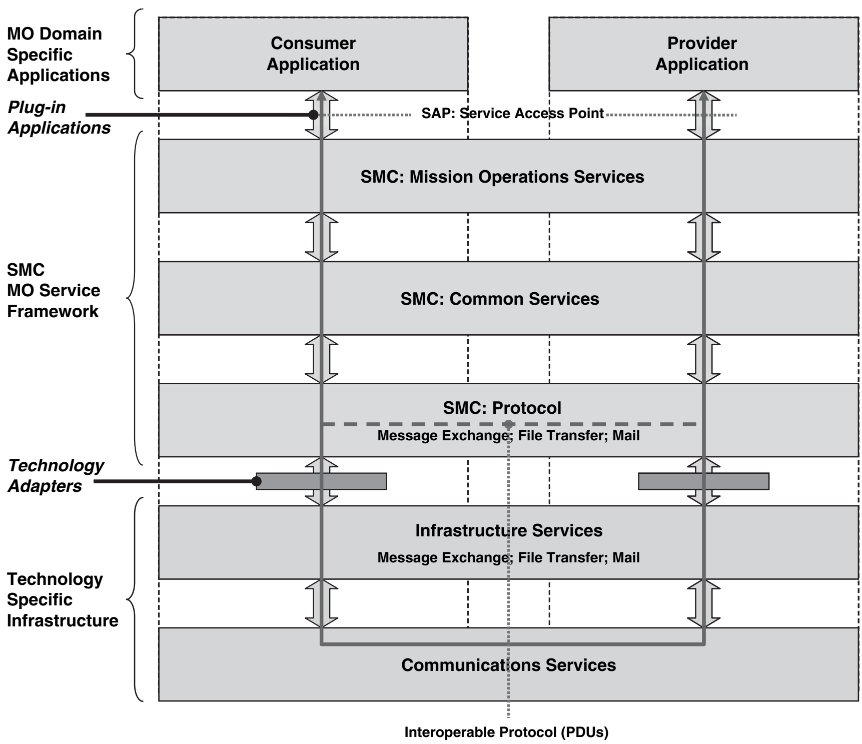

Fig. 3 Overview of SMC mission operations service framework.

for communications between distributed applications. This isolates compliant software applications both from each other and the underlying communications technology. Applications may be plugged in to the service framework through standardized service access points (SAP), one for each end-to-end service. The SAP is defined in terms of a platform independent model (PIM). For any given service, this offers two compatible interfaces that support the service consumer and service provider applications, respectively.

This approach is consistent with the Object Management Group's (OMG) model-driven architecture (MDA) approach and is capable of being fully modeled using the Unified Modeling Language (UML). Tools exist to support this approach that support and even automate the process of generating first the platform-specific model (PSM) for a given deployment technology and then the associated program code. When the SAP is bound to a specific deployment technology (e.g., programming language), it is cast as an application programmers' interface (API) specific to that technology. This API forms the interface to a reusable library that implements the MO service framework, and is called directly by the application software.

The MO service framework itself is also independent of the underlying technology-specific infrastructure. Each deployment technology requires an adapter that allows the framework to be deployed over it. This abstraction of the MO service framework from the deployment infrastructure implementation means that the entire framework can be migrated from one deployment technology to another without modification to the domain-specific applications themselves. It also allows bridging between different technologies, where these are suited to particular communications environments, or to accommodate different implementation choices between agencies.

B. SMC MO Service Framework Layers

The SMC MO framework has three layers, as illustrated in Fig. 4 and in the following sections.

1. SMC Mission Operations Services

This layer provides the end-to-end services that are exposed to mission operations applications. Multiple services have been identified, each corresponding to a particular class of information that is exchanged between mission operations applications and include (non-exhaustive): core SMC, planning, scheduling, automation, flight dynamics, time management, and onboard software management.

Each service is defined in terms of an information model that defines a set of service objects that are shared by providers and consumers of the service. Examples of such service objects are status *parameters*, control *actions*, and notification *alerts*. These constitute the basic elements of the core SMC service. Other services concern specialized information such as orbit vectors, schedules, planning requests, and software images. In addition to definition of the static information model, the service defines the interactions required between service provider and consumer to allow 1) the service consumer to observe the status of objects through a flow of *event* messages, and 2) the service consumer to invoke *operations* upon the objects.

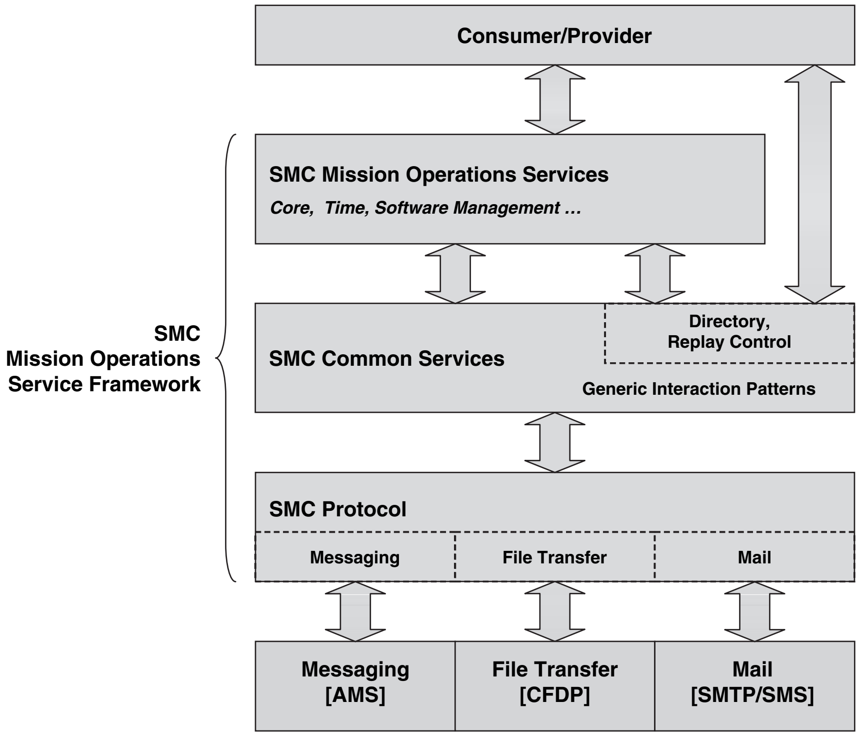

Fig. 4 SMC service framework layers.

The service definition specifies the structure of the information objects exposed at a particular service interface. In most cases, however, each deployment (or instantiation) of a service will also require service configuration data that detail the actual service objects that exist for that service instance. For example, the core SMC service may define what parameters, actions, and alerts are, but it is the associated service configuration data that specify the set of parameters, actions, and alerts that exist for a particular spacecraft.

2. *SMC Common Services*

This layer organizes in a single place all common and generic service elements. In addition to the horizontal layering of services, the SMC MO service framework can be broken down into a number of vertical elements. Some of these elements or subservices are common to all MO services. An example of this is the *directory service* that allows consumers to locate providers of the services they require. These common service elements are directly exposed to applications.

In analyzing the requirements for several potential end-to-end mission operations services, it became apparent that there is a lot of commonality between

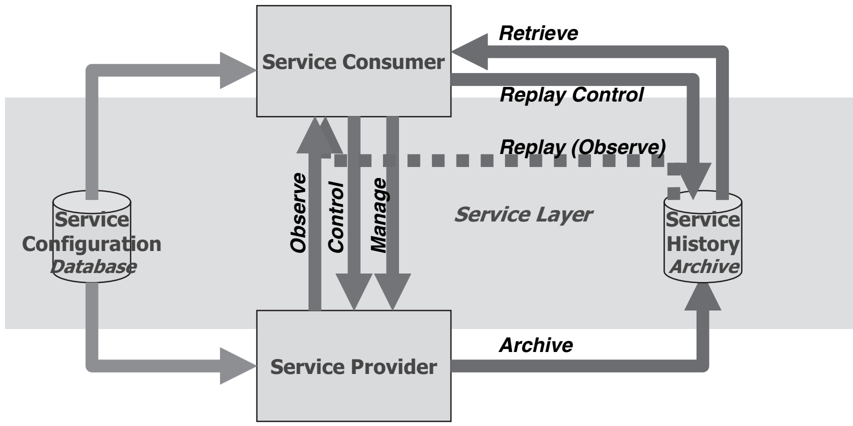

Fig. 5 Generic interaction pattern for mission operations services.

services. While the definition of objects, events, and operations are specific to the service, multiple services can be based on the same fundamental interaction pattern. ~~This~~ includes abstract service elements such as 1) observe, interface in which the service consumer registers for interest in particular service objects and then receives status update events for those objects (e.g., obtain parameter, action, or schedule status); 2) control, interface in which the service consumer can initiate operations on objects (e.g., set parameter, send command, load schedule); and 3) manage, interface in which the service consumer can modify the behavior or processing of the service provider.

Layering of the mission operations services over these generic interaction mechanisms simplifies the task of building adapters to the underlying communications technology, as only one adapter is required to support all mission operations services. In addition to these direct interfaces, there is scope to provide generic infrastructure support for recording and retrieving service history as multiple services are based on the same generic observe interface and also for the distribution of service configuration data.

3. SMC Protocol

The previous layers have been concerned with end-to-end interaction between service provider and consumer. Emphasis has been on the definition of service access points and the standardization of the vertical communication between the layers. Given that the implementation of the service framework itself may differ between agencies and systems, it is critical for these infrastructures to be able to interoperate such that there is standardization of the messages [or protocol data units (PDU)] that pass between provider and consumer.

The SMC protocol layer provides this horizontal standardization between interoperable implementations of the MO service framework. The communications protocol stack beneath the MO service framework must be equivalent on both sides of the interface. Within the MO service framework itself, it is the protocol layer that ensures interoperability, as the bindings between it and the higher layers are standardized. The protocol layer allows for three fundamental communications methods: 1) messaging (which could be implemented over AMS), 2) file transfer (which could be implemented over CFDP), and 3) mail.

It provides a thin layer within the service framework that allows separate technology adapters to be implemented for the underlying communications protocols used for each of these communications methods. Emphasis is placed on the messaging protocol, which is based on the target-controller pattern for monitoring and control. In the context of the SMC MO service framework, the controller is equivalent to the service consumer and the target is equivalent to the service provider. A controller can be a ground control system, an onboard data handling subsystem, or a processor of a payload/subsystem. A target can be a device, a subsystem (ground based or onboard), or even an entire spacecraft. This target-controller pattern can be applied recursively.

There is a standard set of operations that the SMC common protocol provides, that may be used to transfer directives from any controller to any target and similarly to transfer reports from any target to any controller. This standard pattern of interaction may be used to implement any of the following standard operations: 1) trigger execution of target, 2) send directive to target, 3) read state of target, 4) send indication to controller, and 5) send event to controller.

V. Summary: Working Group Status and Outlook

At the time of writing of this chapter, the SMC WG has achieved 1) publication of the SMC Green Book [1]; 2) advanced draft standards for SMC protocol, SMC common service, and SMC core service; and 3) initial draft standards for SMC time service, SMC remote software management, and SMC automation service. Additional information on the work-in-progress could be obtained at the web site of the SMC WG [2].

Additionally, the WG ~~is currently working on~~ a prototype that implements the advance standards previously discussed for validation purposes. An initial version of the prototype was successfully demonstrated in June 2006 and an upgraded version ~~will be presented~~ in January 2007. **Q7**

References

[1] *Mission Operations Services Concept*, Consultative Committee for Space Data Systems, CCSDS 520.0-G-2, Green Book, Issue 2, Washington, DC, Aug. 2006; also *http://public.ccsds.org/publications/archive/520x0g2.pdf*. **Q8, Q9**

[2] Spacecraft Monitor and Control WG Public Area, *http://public.ccsds.org/sites/cwe/moims-sc*.