Report Concerning Space Data System Standards

# Telerobotic Operations

Draft Informational Report

CCSDS 000.0-G-0

Draft Green Book
October 2014

Shortcuts: Document Control • Work List • Contents • Figures and Tables • Lexicon

## *Authority*

| | |
|---|---|
| Issue: | Draft Green Book, Issue 0 |
| Date: | October 2014 |
| Location: | Not Applicable |

**WHEN THIS INFORMATIONAL REPORT IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:**

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in the Procedures Manual for the Consultative Committee for Space Data Systems.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

## *Foreword*

> Agencies agree that human space exploration will be most successful as an international endeavor because there are many challenges to preparing for these missions and because of the significant social, intellectual, and economic benefits to people on Earth.
>
> – *The Global Exploration Roadmap*

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Report is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site: https://www.ccsds.org/.

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page 2.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

<u>Member Agencies</u>

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- Federal Space Agency (FSA)/Russian Federation.
- UK Space Agency/United Kingdom.

<u>Observer Agencies</u>

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.

- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

## *Document Control*

| **Document** | **Title and Issue** | **Date** | **Status** |
|---|---|---|---|
| CCSDS 000.0-G-0 | Telerobotic Operations, Draft Informational Report, Issue 0 | September 2014 | Current Draft |

## *Work List*

The Work List is an unprioritized list of items needing attention from contributors to the Telerobotic Operations Green Book. This section of the document will be deleted before publication.

- Clean up this list, and remind each listed person — by @mention comment — of their intention to contribute content in that area. Follow up on the @mention with an email to their work address. [David]
- When the Green Book has reached the penultimate draft of Issue 1, prune the Lexicon of unreferenced terms, and mark each remaining term as a CCSDS Term, a Term of Art or other type as appropriate. Check the remaining CCSDS Terms against the definitions from the SANA Registry of CCSDS Terms to ensure accuracy. [TBD]
- TODO: Complete the Definitions, Nomenclature and Conventions section patterned after the CSTS Red Book to provide up-front definition of the most important key words. [DSM]
- Add material explaining André’s Layer, Component, Service model. [André]
- Write explanation of Telerobotic Operations services as MO Services. [Mehran]
- Complete Dissimilar, Redundant Robots scenario. [Philippe]
- Complete Shared Autonomous Control scenario. [André or Thomas]
- Complete Multiple Control Points for Single Robot scenario. [David]
- Complete Human-Robot Collaboration scenario. [David]
- Complete Sequencer vs On-Board Queue comparison. [Mehran]
- Expand the Data Product service description or remove. [David]
- Complete conversion of the Command Service to prose. [David]
- Write the Audio Streaming section in Open Questions. [TBD]
- Write the Virtual Telepresence section in Open Questions. [David]
- Complete the list of Other Industry Standards. [David]
- TODO: Write CLARAty description. [David]
- TODO: Finish describing the Robot Operating System. [David]
- TODO: Update the *Mission Operations Common Services* document number

when next issued.

- TODO: Determine what items can be left unspecified in the Green Book, process for recording “unfinished” items, potential for designating areas of standardization for subsequent editions of a Blue Book. [Lindolfo]
- TODO: Determine if the Mission Operations Common Services [10] Directory Service satisfies the requirements of the Discovery Service. [Mehran]
- TODO: Determine if a specialized Voice and Audio Streaming service is required for Telerobotic Operations. [Osvaldo]

*Contents*

## *Figures and Tables*

# 1. Introduction

## 1.1. Purpose and Scope

This Green Book contains overview or descriptive material, supporting analysis, requirements, descriptions of use, scenarios, etc., that will help bound the scope of the subsequent Blue Book.

The contents of this this Green Book is limited, generally, to information that adds value to the process of developing our Telerobotic Operations standard. It was decided at our inaugural Telerobotic Operations Working Group meeting that the main content of the Green Book will be technical in nature. The following classes of non-technical data are specifically and purposefully not included in this document:

- **Administrative**: The Green Book will not include “management” or “collaboration” information; descriptions of schedules, collaboration tools, etc., need to find another home.
- **Advocacy**: Although additional materials (presentations, movies, etc.) from members describing current technical developments in collaborative telerobotics and standards are valuable, they will not be included in the Green Book.

Finally, we must be able to clearly show how the information presented in this Green Book relates to the MOIMS-TEL charter to develop standards that support the safe, collaborative operation of mixed teams of human and robotic assets in the exploration of space.

This document is being collaboratively edited using Google Drive. Here are some best practices for managing the online, realtime, collaborative editing process:

1. Feel free to “add value” to the document via direct edits of the text. Adding value includes correcting spelling and grammatical errors, correcting flow of information, and adding and correcting non-controversial text.
2. Comment freely on the text using the Insert comment tool in the toolbar. Select the text you wish to comment on, press the Insert comment tool, and enter your comment text. Your comment will receive replies from other members of the editing community and will eventually be resolved. You can learn more about discussion comments in the Google Docs Blog entry entitled “Introducing discussions in Google Docs.” Once a comment is marked as Resolved, you can

view the entire discussion, and even reopen the Comment, by clicking the Comments button at the upper right corner of the document window in your web browser.

3. If you wish to perform more extensive editing of paragraphs, sentences, etc., feel free to use blue colored text (as we have done in this list) and/or include editorial comments in square brackets [like this].

### 1.2. Document Structure

Following the introductory material, this document is organized into these main sections:

- Section 2: Introduction to Telerobotic Operations
- Section 3: Definition of Telerobotic Operations
- Section 4: Overview of the Mission Operations Service Framework
- Section 5: Document Roadmap
- Section 6: Telerobotic Operations Use Cases
- Section 7: Other Industry Standards

### 1.3. References

The following documents are referenced in this Report. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Report are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS documents.

[1] *Mission Operations Message Abstraction Layer*. Recommended Standard, Blue Book, Issue 2, CCSDS 521.0-B-2.. CCSDS: Washington, D.C., March 2013.

[2] *The Global Exploration Roadmap*. International Space Exploration Coordination Group, September 2011.

[3] *Mission Operations Services Concept*. Informational Report, Green Book, Issue 3, CCSDS 520.0-G-3. CCSDS: Washington, D.C., December 2010.

[4] *A Strawman Gap Analysis for US-TRSIG*. Scott Burleigh. Pasadena, California: NASA Jet Propulsion Laboratory, April 14, 2011. (Chart)

[5] *Voice and Audio Communications*. Proposed Practice, White Book, Issue 0,

CCSDS 766.2-W-0. CCSDS: Washington, D.C., November 2013.

[6] *Motion Imagery and Application*. Informational Report, Green Book, Issue 1, CCSDS 706.1-G-1. CCSDS: Washington, D.C., November 2010.

[7] *Mission Operations Reference Model*. Recommended Practice, Magenta Book. Issue 1, CCSDS 520.1-M-1. CCSDS: Washington, D.C., July 2010.

[8] *Mission Operations Monitor and Control Services*. Draft Recommended Standard, Red Book, Issue 3, CCSDS 522.1-R-3. CCSDS: Washington, D.C., March 2014.

[9] *Mission Operations Common Object Model*. Recommended Standard, Blue Book, Issue 1, CCSDS 521.1-B-1. CCSDS: Washington, D.C., February 2014.

[10] *Mission Operations Common Services*. Draft Recommended Standard, Red Book, Issue 2 Draft 1, CCSDS 522.1-R-2 Draft 1. CCSDS: Washington, D.C., August 2011.

TODO: Update the *Mission Operations Common Services* document number when next issued.

[11] *CCSDS File Delivery Protocol*. Recommended Standard, Blue Book, Issue 4, CCSDS 727.0-B-4. CCSDS: Washington, D.C., January 2007.

[12] *Space Data Link Security Protocol*. Draft Recommended Standard, Red Book, Issue 2, CCSDS 355.0-R-2. CCSDS: Washington, D.C., February 2012.

[13] Security Architecture for Space Data Systems. Recommended Practice, Magenta Book, Issue 1, CCSDS 351.0-M-1. CCSDS: Washington, D.C., November 2012.

[14] SensorML: Model and XML Encoding Standard. OGC 12-000. Open Geospatial Consortium: Wayland, Massachusetts, 2014.

[15] Reference Model for Service Oriented Architecture 1.0. SOA-RM. OASIS Open Standard: Burlington, Massachusetts, 2006.

## 1.4. Definitions of Acronyms

For a complete list of official CCSDS Abbreviations, consult the SANA CCSDS Abbreviations registry.

| | |
|---|---|
| AMS | CCSDS Asynchronous Message Service |
| API | Application Programmers' Interface |
| CCSDS | Consultative Committee for Space Data Systems |
| CFDP | CCSDS File Delivery Protocol |
| JAUS | Joint Architecture for Unmanned Systems |
| MAL | Message Abstraction Layer |
| MO | Mission Operations |
| RAPID | Robot Application Programming Interface Delegate |
| SANA | Space Assigned Numbers Authority |
| SM&C | CCSDS Spacecraft Monitor & Control |
| SOA | Service Oriented Architecture |

## 1.5. Definitions, Nomenclature and Conventions

This Information Report makes use of a number of terms defined in reference [3]. The use of those terms in this Informational Report shall be understood in a generic sense, i.e., in the sense that those terms are generally applicable to technologies that provide for the telerobotic operation of real systems. Those terms are:

- Application Programmers' Interface
- Operation
- Service Consumer
- Service Provider

TODO: Complete the Definitions, Nomenclature and Conventions section patterned after the CSTS Red Book to provide up-front definition of the most important key words. [DSM]

# 2. Introduction to Telerobotic Operations

Extending human presence and capability into space will require collaboration between mixed teams of human and robotic assets; robots will be tasked with the “dull, dirty, or dangerous” work that is not sensible or necessary for humans to perform. Current space telerobotic operations concepts do not scale well beyond the “one operations team, one robot” approach, resulting in an inability to efficiently scale operations to multi-robot teams. By including humans in mixed human-robot teams, we add the additional complexity of human safety concerns, which generally further complicate the telerobotic operations concept by imposing constraints on the robotic elements such as keep-out zones and movement speed and force limitations. Finally, the most significant barrier to international cooperation in space telerobotics is the basic inability to intercommunicate; there is no common language by which a diverse set of human and robotic collaborators can share information for the purpose of achieving a common goal.

## 2.1. Summary of Telerobotic Operations

This section contains a description of the Mission Operations Services Concept and the Service Oriented Architecture upon which Telerobotic Operations is based. The concept layers depicted in Figure 1 are based on the notional model of telerobotic operations described in this section.

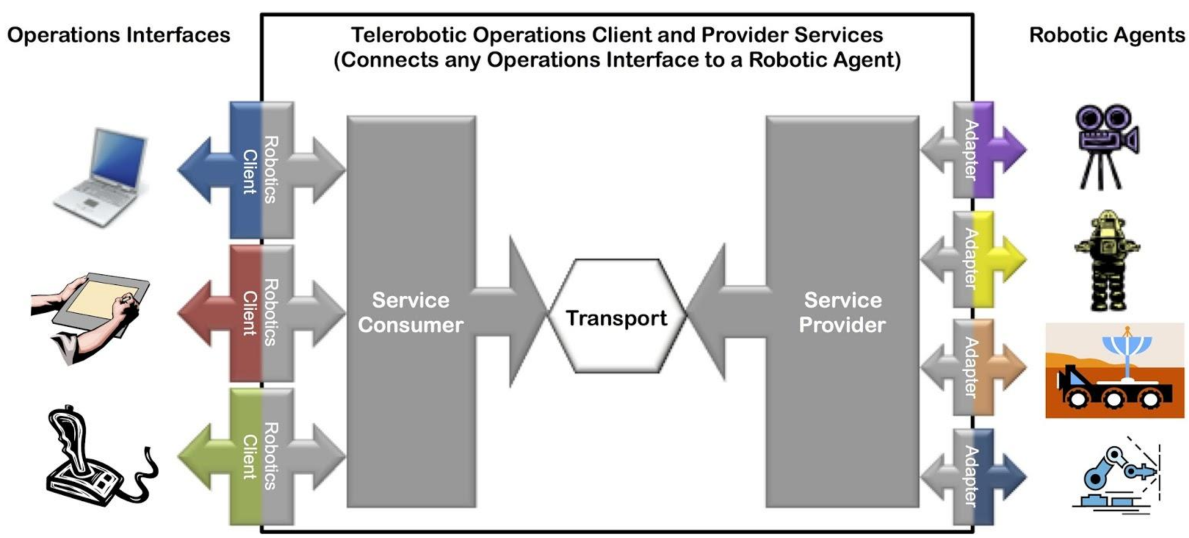


**Figure 1. Telerobotic end-to-end model.**

The list below describes some of overall Telerobotic behaviors and service dependencies which will be described in this document. Some of the listed services are the subject of other CCSDS standards; standards upon which Telerobotic Operations will depend, but will not define. A complete list of services can be found in Section 3.2, Telerobotic Operations Services.

1. Robotic devices (on the right) publish telemetry (including, potentially, streaming audio and/or video) to one or more subscribers, which may include both human operators and other robotic devices.
2. Human operators (on the left) send operational command data privately to selected specific robotic devices.
3. Under some conditions, human operators might publish information broadly to one or more robotic devices. Such information might include environmental data (e.g., updated maps) or contextual data governing local autonomous behavior (e.g., goals, changes in policy, revised alarm limits, etc.).
4. Human operators may need to communicate among themselves, by text message and/or by streaming audio [5] and/or video [6]. This communication may be either private (directed from one human operator to another) or public (directed from one human operator to a set of others, where the set of receiving operators may be self-selected – i.e., subscribers – or may be selected by the sender).
5. Some subset of this communication may require authorization, authentication, and/or confidentiality.
6. Some subset of this communication must be assured by acknowledgment and retransmission procedures – but some must be delivered in the order in which it was transmitted, without delay, and therefore should not be subject to acknowledgment and retransmission.
7. Both interactive (real-time, including haptics and force-feedback) operation of proximate connected robotic devices and policy-driven (autonomous) operation of remote or temporarily disconnected devices must be accommodated by Telerobotic Operations. Services must be robust to a wide range of time-delayed and disrupted communications.
8. Centralization of the monitor and control function is possible, depending upon the architecture of the communication network and the needs of the exploration system. A centralized system would integrate one or more of the left hand or right hand elements into more capable services.

## 2.2. Overview of Telerobotic Operations Layers

Telerobotic Operations utilizes a Service Oriented Architecture defined by the CCSDS Mission Operations Spacecraft Monitor and Control Working Group. Telerobotic

Operations will act as a services layer just below the Device specific displays and machine/device layer at either end of the telerobotic operations as shown in Figure 1.

The CCSDS MO layer will provide end to end communication and Monitoring and Control services to specific Telerobotic services such as Discovery, Robot Command, Robot Telemetry etc. The SM&C services are part of the overall Mission Operations Services that will be described in the next section.

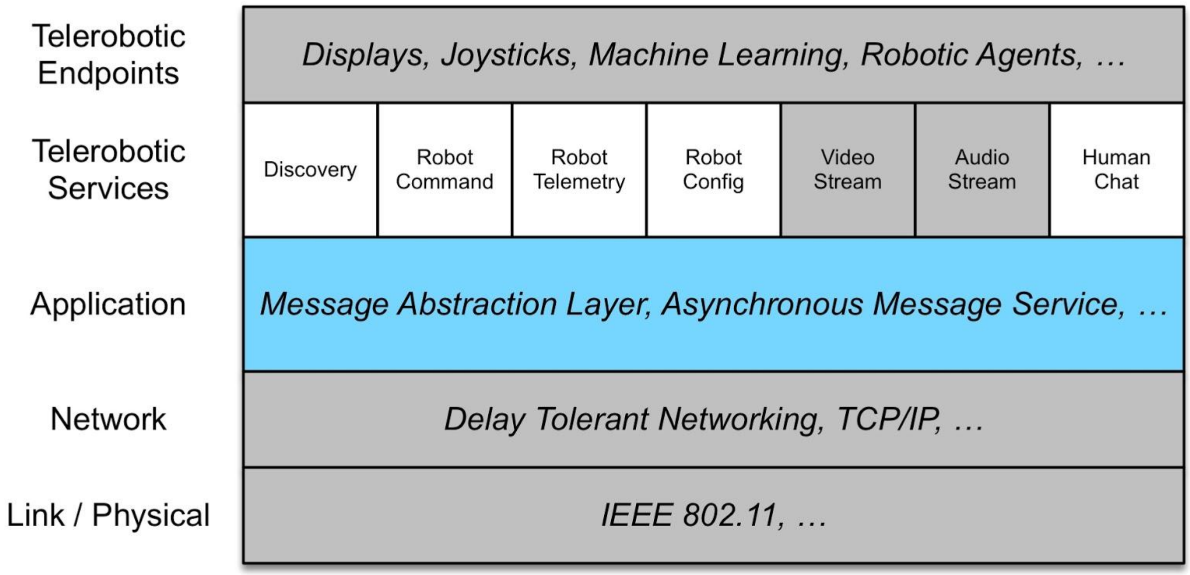


**Figure 2. A strawman gap analysis.**

In Figure 2, protocols for which well-established standards exist are shown in gray. Protocols that are being standardized within other CCSDS Working Groups are shown in blue. Protocols shown in white are being standardized by the CCSDS Telerobotic Operations Working Group. The precise structure of the network stack is dependent upon the deployment or implementation; conceptual layers can be merged for performance reasons or other constraints. This will likely be the case for deployments on-board spacecraft and in a robotic system with limited computing resources.

[AS/MS/LM] Addition, discussion, etc. of André's Layer, Component, Service model.

### 2.3. Relationship to the Mission Operations Services Concept

Telerobotic Operations is, in many ways, an extension of the Mission Operations Services Concept [3]; the reader who is unfamiliar with the latter subject is directed to read the overview material presented in Overview of the Mission Operations Service

Framework.

Each Telerobotic Operations service provides a set of well-defined capabilities through a standardized service contract (i.e., the interface, specified in the subsequent Blue Book). The Telerobotic Operations service contracts, shall be specified in an implementation and communication technology agnostic manner. For example, a Motion Control service would provide the capability to *Move* a robotic asset to a particular position (absolute or relative), without making any assumption about what programming language is used by the Service Provider or Service Consumer nor making any assumptions about what communication technology would be adopted in a particular deployment scenario to establish the link between the Service Consumer and Service Provider.

Each capability is specified in form of an abstract operation, which itself is defined by a set of exchanged messages between the Service Consumer and Service Provider, following a prescribed interaction pattern. The messages exchanged between the Service Consumer and the Service Provider, would contain both data (message body) and meta-data (message header). The data part of all messages exchanged between the Service Consumer and provider for all capabilities of a service, compose together the data or information model of a service.

The meta-data part of the messages (the header) captures usually the non-functional aspects of interactions between the Service Consumer and the Service Provider, such as the aspects related to:

- Addressing (how to reach the Service Consumer / provider)
- Service Taxonomy (the domain and the name and version of the service)
- Transaction management (how to correlate messages in asynchronous interaction patterns)
- Security
- Quality of Service (Service Level Agreements),
- etc.

In a concrete deployment scenario, the Service Consumer and Service Provider must implement the abstract (i.e. technology independent) service contract, in a concrete technology (e.g. a programming language such as Java or C, ADA, C++ or .NET). In the selected programming language the abstract service contract is typically realized through a language-specific Application Programmers' Interface (API), an interface definitions in the form of operations or methods and data types.

In addition, the service consumer and Service Provider will use a concrete communication technology for the exchange of messages at run-time, which is suitable for that particular deployment scenario (e.g. pure tcp/ip, or http over tcp/ip or SOAP over http over tcp/ip or CCSDS Space Packet binding over AMS over DTN over…).

Here we can now put the CCSDS MO Framework in the context, as by specifying the Telerobotics services as MO services (i.e. in terms of MAL), the MO Framework does the work of abstraction for us. [I will complete this section later]

[Mehran is working on this text.]

# 3. Definition of Telerobotic Operations

In this section, we document our standard by describing the core of the information interface, processing and access methods to be defined in the subsequent Blue Book. In this section, we describe these interface elements at the informational level, leaving the formal specification of the corresponding service interfaces — in terms of operations signatures, interaction patterns, message and data formats as well as error and exception handling — for the Blue Book.

## 3.1. Telerobotic Operations Services

As we intend to specify generic, interoperable Telerobotic Operations Services, we structure the core of our standard into the main elements of a Service Oriented Architecture: Services, Operations, Messages and Data Types:

- Services shall be expressed as a collection of Operations; each Operation provides a particular telerobotic capability that is relevant for interoperable collaboration.
- Operations shall be expressed as a specific well-defined pattern of exchanged Messages between the two or more parties involved (i.e. the Service Provider and Service Consumers), in order to achieve the corresponding collaboration capability.
- Messages shall be specified as a collection of Data Types exchanged for each interaction step.
- Data Types shall provide the formal specification for the information exchanged through messages.

Telerobotic Operations Services are defined, in part, by the information they carry between members participating in collaborative telerobotic operations, and is roughly divided into two categories: information conveyed as part of a commanding operation and information conveyed as part of a monitoring operation. We further divide our Services into two broad categories: those that are specific to telerobotics, and those that have a broader applicability. Those with more broad applicability are likely to be the subject of other standards definitions.

### 3.1.1. Telerobotic Operations Scenarios

In this section, we highlight elements of human and robotic exploration activities and how they might utilize the Telerobotic Operations services described in this document. The scenario descriptions include both functional elements (for instance, keep-out zones, which would utilize the Frame Store Service to monitor collaborator locations and warn of incursions) and performance elements (for instance, data rates and communication latency). Scenarios that reference advanced capabilities, such as optical communications, also include a roadmap for qualifying the technology for mission use. In developing these operations scenarios, an attempt has been made to ensure that each of the services described in this document is referenced in at least one operations scenario.

The operations scenarios listed in this section are not exhaustive; many more potential operations scenarios might be described with additional time and space, and in support of future exploration plans and architectures developed by member Agencies. The reader is encouraged to consult Agency and international roadmaps, such as the Global Exploration Roadmap [2], to establish an alignment between the scenarios described in this section, and the larger context described in the exploration roadmaps.

For style and content consistency, we need a set of guidelines for scenario content. What aspects of the scenarios should be stressed:

- Collaboration needs (inter-Agency, inter-Center, Human / Robot)
    - Cross-Agency Support
        - Testing (Field or Laboratory)
        - Simulation Services
    - Emergency Support
    - Nominal Operations Support
    - Research and Technology Development
    - General Service Level Agreements
        - Time and Resource Sharing, etc.
- Suitability
    - When is it **not** appropriate to use Telerobotics Operations; possibly because of degraded network capability, etc.
- Time delay regimes
    - 850ms vs 2s vs 50s vs 20m
        - Roughly: Force reflection stability boundary, Lunar, NEO and Mars
    - Communications Infrastructure Service Level Agreements
    - Time-of-Flight Restrictions
- Number and distribution of operators and agents
- Mission destinations (Earth orbital, Cis-Lunar, etc.)
- 5-, 10-, 25-year development schedules

#### 3.1.1.1. Motion Imagery Scenario

Video and motion imagery systems, particularly when designed for external spacecraft applications are, in essence, simple robotic systems. Typical systems would likely include a camera with a zoom lens, mounted onto a platform that can be panned (moved left to right) or tilted (pointed up or down). The camera itself will likely have several functions that need to be controlled and monitored, such as iris or exposure, white balance, and other modes such as frame rate and resolution. The lens would need to be focused and the focal distance changed (zooming in or out). The mount, or pan-tilt unit, would also need to be controlled or automated. Metadata from the camera system would be very important, enabling viewers down-stream to verify the camera's position in three-dimensional space, or providing indicators for where the camera is pointed on an x- and y-axis. Operating temperature and other similar data would also be critical to ensure proper operation.

To date, most video and motion imagery systems flown on spacecraft have utilized unique or proprietary interface controls, software and/or hardware to manipulate the camera system. This proliferation of control approaches leads to inefficiencies, increases costs, and reduces redundancy and interoperability. Development of common services and protocols that could be reused for video and motion imagery systems would be highly desirable as future missions are likely to involve multiple space agencies, private enterprises, or combinations of both.

Among the challenges of incorporating video and other motion imagery data in Telerobotic Operations is that these data sources consume a large portion of the bandwidth available in space communications links. Advances in motion imagery compression will improve the situation somewhat, but those advances will likely be offset by the increased use of virtual telepresence systems for operations that include immersive technologies such as three-dimensional video.

The subject of Virtual Telepresence as part of the Telerobotic Operations standard is an open question, discussed in Section 3.4.4, Virtual Telepresence.

#### 3.1.1.2. Dissimilar, Redundant Robots

[Philippe Schoonejans]

In future robotics exploration scenarios — but also inside and outside orbital stations in LEO, near Moon or Mars — several robots may be working together. In the foreseeable future, there will always be just a few people around, with a very high workload. Robots who will be able to perform tasks designed for humans will be very useful to take away some of this workload, but will also have a role in performing dangerous and/or repetitive tasks. Robots are getting more and more autonomous, but still some level of monitoring by humans will be foreseen and for some tasks also tele-operation will be useful.

[DSM] Extrapolate from the current situation on ISS — with Robonaut 2 and SPHERES performing housekeeping chores — to future station-keeping activities before, during and after human crew occupy a deep-space habitat (Mars transit?).

Especially when going beyond LEO, any failure in such a robotic system will be a major issue, as on-board repair and replacement will be impossible or at least very costly. The reliability on mission level will be significantly increased if the robots will be of dissimilar design, to minimise the risk of common cause failures due to unforeseen design flaws or wear. This provides also an opportunity for several space agencies to develop their robotic capabilities by building and launching a (humanoid) robot to LEO, Moon or Mars.

Hence, bringing together robots from several space agencies will increase the number of robots available for work in space as well as provide dissimilar redundancy for single robot operations.

An example scenario could be the use of robots like NASA's robonaut, ESA's Eurobot, Russia's SAR-401 on LEO or cis-lunar space stations.

The same dissimilar redundancy rationale could, although to a lesser extent, also be given for the reliability of the robots' monitoring and control stations, who, in case of master/slave control systems, could also be of significant complexity.

An important enabler for above scenarios would be if workstations and robots would follow standardised interfaces and operations concepts and therefore be interoperable. Robots could be each other's backup, possibly even hot backup. This standard should make a contribution to such a future goal. [Cross-Agency support.]

#### 3.1.1.3. Shared Autonomous Control

[André Schiele] Real-time bilateral control, autonomous assisted control, and some full autonomy for a Centaur-type robotic system. Address goal-oriented commanding, partitioning of autonomy between flight and ground elements, and supervision of autonomous behaviors.

* begin edit tkrueger

Different levels of autonomy due to different delays and quality of the the connection.

Low latency (~100 ms) - real-time control

- Orbiter around the planet with the control master
- Robot on the surface
- Line of sight communication
- Specific: S-Band from ISS to a specific ground stations

Latency of 1-4 seconds - shared control / model mediated control

- Orbiter around the planet with the control master
- Robot on the surface
- Communication via relay via satellites or ground stations
- Specific: ISS to ground via

Very High latency or temporary loss of connection or/and combination of automatable tasks
Full autonomy
//* end edit tkrueger

Operations scenarios: Robotic surgery / telemedicine. Disruption scenarios.

How to guarantee the safety of a telemedicine system (surgery) in the presence of link degradation (what happens when you're in the middle of an incision, and your comm link disappears?). How do we ensure that all such systems adhere to a consistent set of safety guidelines, and express a consistent set of requirements on the underlying communications systems? This scenario might be a good place to discuss the "suitability service." [DSM]

Safe operation of telesurgery systems require bilateral control, which blah, blah, blah.

#### 3.1.1.4. Control of Multiple Agents Simultaneously

[Philippe Schoonejans]

As indicated in 3.1.1.2, scenarios are anticipated where robots and people will work together. In principle, one person can be leading several robots. The robots are autonomous to a certain level. They do not continuously need guidance, monitoring and control, but they need it depending on the task they are executing at any moment in time. The consequence of such a scenario for the system architecture is that it would be ideal for a single workstation to be able to control multiple robots simultaneously, be it from Earth, from a zero-g orbiter or space station or from a surface base.

[DSM] Refer to the listing at the beginning of this section for additional descriptive elements that can be added to the above paragraph. What mission concepts are being considered that might require this type of multi-agent, hierarchical control? What development timelines have been proposed? Etc.

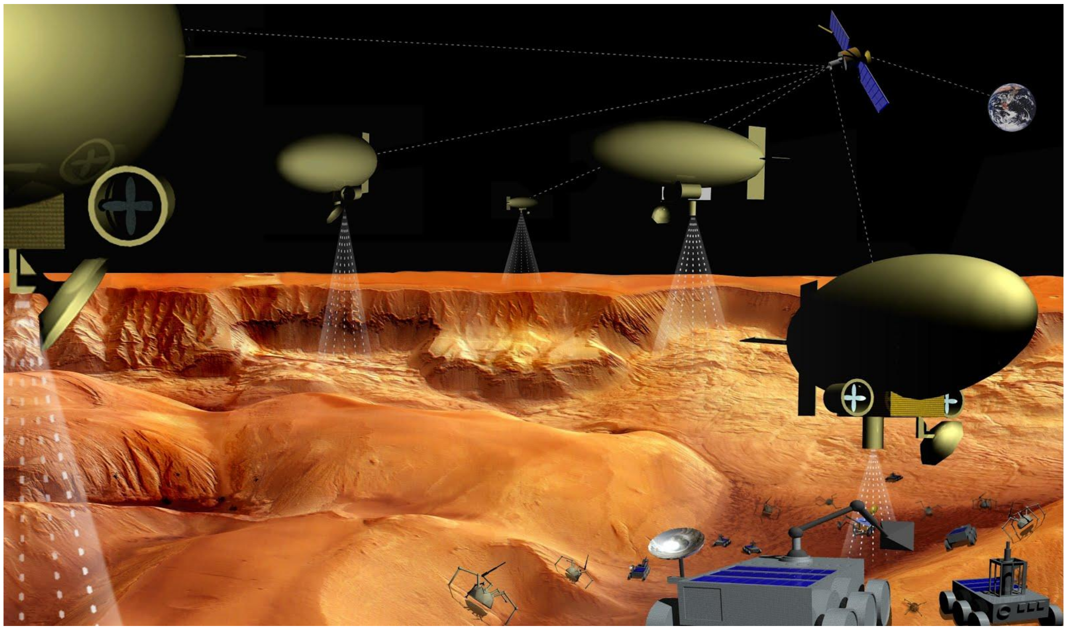

**Figure 3. Artist's concept of orbiter, blimps, rovers and robots working together.** *(Image Courtesy NASA/JPL-Caltech)*

This would be enabled by standardisation of interfaces to achieve interoperability of robots.

Swarm Robots Notes

- Timeframe?
- Destinations? (Mars is mentioned)
- Mission concepts.
- Development schedule.

Swarm robotics is a relatively new technique that applies systems of multiple identical or nearly identical robot agents to the solution of a complex problem. In field robotics, such as the exploration of the **Martian surface**, a system of similar robots can cover a larger percentage of unknown terrain in a shorter duration, increasing the amount of time spent exploring any single target, and improving the odds for making the chance discovery. In orbital robotics, swarm robotics often appear in the literature as networks of formation flyers. As scientific platforms, formation flyers can be used to increase the effective baseline of an **observation platform**, such as in synthetic aperture radars or high-resolution remote sensing platforms. **Formation-flying** technologies are also being applied to fractionated spacecraft, where the functions performed by a single monolithic spacecraft are partitioned to a swarm of independent spacecraft acting as functional modules of a system. Among the benefits of swarm techniques is the reduction in risk from the loss of any single agent, allowing the swarm system to be used to accept more risk and operate in more challenging environments.

In many cases, individual elements of a swarm are tightly linked to other elements through common spacecraft design, relative positioning requirements and data system interfaces. Currently, swarm systems are generally produced by a single provider and utilize a custom flight / ground interface and control methodology. By applying standards in Telerobotic Operations services, we can open up the swarm system to multiple agent providers, increase **cross-agency support** and enable the control of multiple swarm systems through standard ground control station technology.

#### 3.1.1.5. Multiple Control Points for Single Robot

[David Mittman] Considering an R2 mounted on Centaur 2, one controller for the base, one for the arms, working simultaneously. Also include the case of control points with different latencies (Earth to Lunar, Cis-Lunar to Lunar, Lunar surface to surface, etc.).

In a “follow the sun” operations scenario, the Transfer of Control service would be used to transfer operator control of robotic assets between mission control centers (USA > ESA > Russia, for example).

#### 3.1.1.6. Mehran Special Scenarios

Multi-Robot System level Infrastructure:

In a simulation infrastructure scenario like METERON, where multiple robotic agents can participate in implementing different robotic mission scenarios, there is a need for a system-wide **simulation** environment as well as for a generic system level monitoring and control infrastructure, into which the participating robotic agents can plug in with their specific robotic control systems (software and hardware).
Each robotic system would come with its own proprietary robotic control system, RCS (hardware and software, e.g. a laptop on which a control software is deployed or a set of machines).

Specifying standardised Robotic Services allows to abstract from the proprietary interfaces and facilitate the integration of various robotic systems into the system-wide simulation and monitoring and control infrastructure.
It is important to emphasise that the role of the Standardised Robotic Services in this context is not to replace the Robot specific RCS but to facilitate its integration into higher level, multi-Robot-supporting infrastructure.

Distributed and collaborative monitoring and control systems :

This is less a scenario than it is an argument in favor of the standard. [DSM]

In a collaborative robotic operations scenario, where two or more robotic agents of more than one entity are engaged in joint operations, each entity would probably use its own monitoring and control system for planning, executing and monitoring the operations of its own robotic system. Standardised Robotic Services would allow the exchange of operational information between the involved entities, without imposing the use of one and the same common robotic monitoring and control system by involved parties. In other words, while each entity can still use its own M&C systems and infrastructure, by implementing the Standardised Robotic Service interfaces, they can receive information about the operations of the other robotic systems from the other M&C systems (e.g. getting TM from other Rovers and visualise it in one's own M&C system, receiving commanding status updates from other M&C systems, receiving communication status from other M&C systems, etc.). Some of the standardised Robotic Services, will facilitate in particular the coordinated operations. For instance if multiple entities are involved in a distributed operations setup, where entity one must execute operation/plan one before entity two can start its operations, the Transfer of Control Service will enable such coordination.

The important aspect to to point out is the role of the envisaged Standardised Robotic Services in exchange of relevant information between distributed and diverse Robotic Monitoring and Control Software Applications.

#### 3.1.1.7. Inter-Agency, Inter-Center Field Testing

International space agencies are planning to expand human and robotic exploration beyond low Earth orbit to destinations including the moon, near-Earth asteroids, and Mars and its moons. To prepare to explore these destinations, Agencies are first conducting analog missions here on Earth. Analog missions are remote field tests in locations that are identified based on their physical similarities to the extreme space environments of a target mission. Working together, engineers and scientists from government agencies, academia and industry gather requirements, develop technologies and test robotic equipment, vehicles, habitats, communications, and power generation and storage. Field tests are designed to evaluate mobility, infrastructure, and effectiveness in the harsh environments.

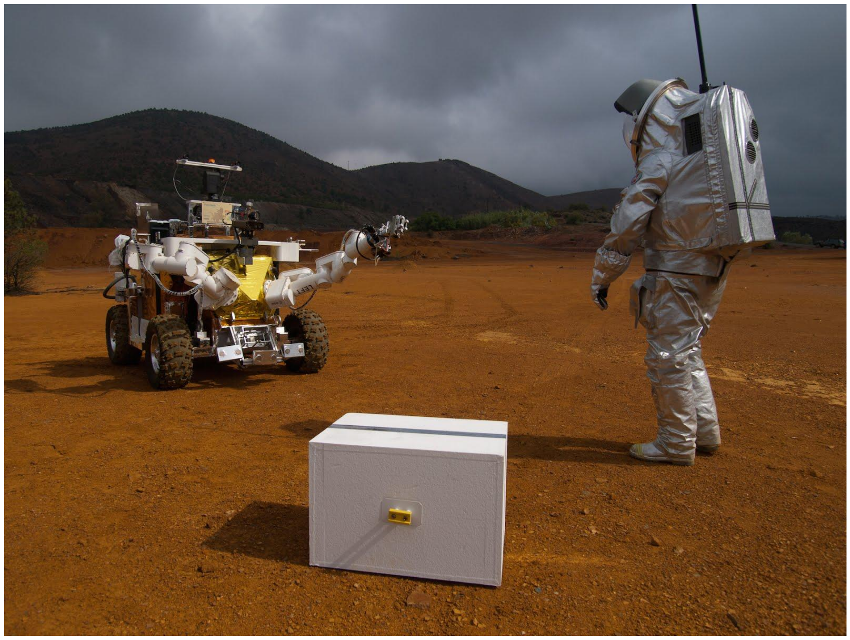

**Figure 4. ESA Rio Tinto field test in Andalusia, Spain.**

Analog missions provide an excellent opportunity to perform interoperability testing of assets from multiple Agencies, as well as a representative environment for testing our Telerobotic Operations standard. The harsh environment and remote location of the field test site often present real-world challenges to the coordination and communication required to meet mission objectives. Heat, dust and terrain present localized communication challenges, often resulting in the kinds of disruption that complicates space links; the remote location often involves long backhaul connections to the internet and related communication delays. Communication delay is often injected as a test condition when delays of tens of seconds are required to simulate distances to near-Earth objects and beyond.

To control costs, analog missions incorporate realistically designed elements of the space exploration system only where such fidelity serves a specific analytical need. In some cases, commercial-off-the-shelf components are used as functional equivalents of their space-qualified siblings. For instance, analog missions often use commercial GPS units for localization in place of more expensive space-qualified hardware (in this example, GPS technology is currently Earth-centric). A similar approach can be used in testing our Telerobotic Operations standard in analog missions by simplifying the communications stack and operating the message and service standard over the terrestrial field test network with commercial technology such as RTI DDS over UDP.

Where terrestrial test network infrastructure currently exists to support inter-Agency communication, we can utilize our standard over the commercial internet between Agency development laboratories. Use of existing infrastructure will reduce the costs associated with the independent prototyping and test activities required of CCSDS standards development groups.

In addition to opportunities to test the Telerobotic Operations standard, field testing offers favorable circumstances for testing cross-agency support, including emerging standardization areas such as scheduling and mission planning. Field testing infrastructure can also be modified to support collaborative use of Agency testbed infrastructure, such as air-bearing floors, reactive robot bases -- macro/micro robotic system, Stewart platforms, etc. The same networks that transport data between field test sites and home Agencies can also be used to connect partner Agencies' testing facilities.

To read more about analog field testing, visit the NASA Desert Research and Technology Studies website or read about the ESA Eurobot test at Rio Tinto.

#### 3.1.1.8. Human-Robot Collaboration

[David Mittman] Lunar outpost construction and maintenance.

#### 3.1.1.9. Scenario X

TODO: Add any additional Scenarios here.

### 3.1.2. Network Architecture

Collaborative telerobotic operations necessarily involves multiple operators and multiple robots operating on a single connected network. While the architecture of the network is not necessarily the subject of this document, certain capabilities of the network can greatly simplify the telerobotic operations that the network supports. Telerobotic Operations will generally reference the CCSDS MO SM&C Operations Domain to help define the mission network operations domains as described in Section 4.5.3.4, Operations Domains.

[DM] Update this section; Operations Domains no longer exists in this document.

Some of the capabilities required include the ability to enable the monitoring of a single Agent's telemetry by multiple Operator stations (and other Agents), the network should support a **one-to-many distribution** model, such as multicast, in addition to point-to-point transmission. Scenarios in which this type of distribution model would be beneficial include a Lunar surface robot being monitored by an adjacent suited astronaut in addition to an IVA support astronaut in a nearby Lunar outpost and Earth-based operators.

[DM] The above paragraph makes oblique reference to DTN.

### 3.1.3. Telerobotic Operations Service Categories

This section of the document defines and categorizes the information exchanged between participants. We relate each of the Telerobotic Operations services in the sections that follow to an existing service provided by CCSDS. The following CCSDS service keys are used:

TODO: This will be "consumed" as these items are discussed elsewhere now.

- Parameter: MO SM&C Parameter (telemetry) Service
- Alert: MO SM&C Monitoring and Control
- Archive: MO SM&C COM
- Motion Control:
- Manipulation Service:
- File Management Service:

Add audio, file exchange / management. File system (data product) management.

The following functions may be required in an end-to-end robotic operational scenario. However some of these functions may not be specific to telerobotics and can be (specialized) reuse of more generic functions. In the latter case we could use scenarios to somehow explain how these generic services (functions) could fit into a telerobotic operations context.

The functions which have an (*) are considered to be more relevant in a collaborative robotic operational scenario:

TODO: This section will be consumed, as they are now written up in other sections.
TODO: Note which section duplicates this content. [LM]

- *(p2p and multicast) Event function (MO SM&C Action, MAL, Alert)

- *Validity and Limit Checking function (MO SM&C Action, Parameter Checks)
- *File Management functions (CCSDS FDP, MO SM&C Action)
- *Archiving functions (MO SM&C COM Archive)
- *Data Distribution Functions (MO SM&C Action, MAL, COM, Alert)
- Planning functions (planning request handling, plan products exchange)
- Memory Management functions
- Reporting Function
- Time Management functions (time synchronisation, time correlation)
- Authorization and Collaborative Commanding Coordination functions (who can do what at which time on what element)

## 3.2. Telerobotic Operations Services

Telerobotic Operations defines common capabilities that greatly increase the ability of Agents to collaborate on common telerobotic tasks. In cases where a service is an extension or subclass of an existing CCSDS service, the CCSDS service is called out in the referenced section.

The following Services are defined in this section:

- Manipulation Service: The Manipulation Service controls the motion of manipulators such as robot arms, booms, sample acquisition devices and cameras.
- Sequencer Service: The Sequencer Service enables time-delayed teleoperation of robotic agents through a synchronous command queue.
- Frame Store Service: The Frame Store Service provides location awareness between robots.
- Asynchronous File Transfer Service: The Asynchronous File Transfer Service provides a robust file delivery mechanism between Agents.
- Access Control Service: The Access Control Service authenticates Agents, authorizes their participation in the Agent network, and ensures the security and reliability of data in transit.
- Transfer of Control Service: The Transfer of Control Service mediates requests by Operators to control Agents by ensuring that requests to transfer control between Agents are handled according to an established policy.
- Task Decomposition Service: The Task Decomposition Service provides traceability from high-level tasks, down through their decomposition into lower-level components — such as operations and actions — to the telemetry produced by those actions.
- Imaging Service: The Imaging Service provides capabilities for capturing images

from Agent-mounted cameras.

- Video Service: The Video Service provides moving images to the Operator as a natural way of representing the dynamic state of the environment in which the Agent is operating.
- Command Service: The Command Service provides a signal that, when sent to an Agent, causes the Agent to perform a function not otherwise covered by the Services described in this document, and for which *some* coordination is required between Agents and/or Operators.
- Direct Command Service: The Direct Command Service provides a signal that, when sent to an Agent, causes the Agent to perform a function not otherwise covered by the Services described in this document, and for which *no* coordination is required between Agents and/or Operators.
- Data Product Service: The Data Product Service provides type-specific access to operationally-relevant data collected by Agents.
- Discovery Service: The Discovery Service provides a method by which to find Agents and their capabilities as they join and leave the network.
- Administrative Service: The Administrative Service provides system-level managerial and organizational functions for test and control. The Administrative Service is likely going to be deleted and its functions transferred elsewhere. For now, consider the Administrative Service to be a "catch-all" service for items with no clear home in one of the other Telerobotic Operations services.
- Location Service: The Location Service establishes the relative or absolute position and velocity of of Agents during surface or free-flight operations.
- Mobility Service: The Mobility Service controls the motion of navigable rovers over a surface.
- Configuration Service: The Configuration Service provides information for describing the existence and state of various components that affect Agent operations.

Create a specific Section to highlight the "services" upon which TEL is dependent? Network, security, file management, etc. "Service/Facility/Environment Dependency"?

Do we need a Service Suitability Service? Is the environment suitable for executing certain telerobotic operations? [PS] This is a secondary consideration, but it can be an important consideration in determining whether or not to perform an action. For instance, NASA does not command its Mars rovers during solar conjunction due to the low likelihood that a command load will be received correctly. Other telerobotic operations may be reserved for situations with better communications coverage or data rates, for instance.

### 3.2.1. Manipulation Service

The Manipulation Service controls the motion of manipulators such as robot arms, booms, sample acquisition devices, cameras and other devices that require articulated control. The Manipulation Service uses the Command Service to effect control over a wide variety of types of manipulators.

Agent manipulation is normally accomplished via discrete commands within the context of having established normal parameters of the manipulation action, such as controller gains, force thresholds, etc. The Command Service provides access to standard mechanisms for executing actions, and for selecting amongst a finite set of well-understood performance specification parameters.

The Manipulation Service can be further broken down into hierarchical parts, each addressing different levels of complexity in manipulation planning, such as Task, Operation, and Action.

TODO: Determine the structure of this planning decomposition, or develop an abstract approach that allows for multiple specifications as part of the TEL standard. Example of Legged robots whose limbs can both provide mobility and manipulation. Six-legged walking robot (ATHLETE, DLR Crawler) -- wheels on limbs, six-legged.

### 3.2.2. Sequencer Service

The Sequencer Service enables time-delayed teleoperation of robotic agents through a synchronous command queue. Sequences…

The Sequencer Service uses the MO Configuration Service [10] to manage a queue of commands to be executed by a designated agent, including Queue Control and Queue Report functions.

The Queue Control operation set includes methods for managing the Sequencer Service's queue of commands, including the insertion and deletion of commands from the queue.

The Queue Report operation set includes methods for providing status messages that reflect the Sequencer Service's current state.

TODO [Mehran]: Compare the requirements for a Telerobotic Operations Sequencer Service to the functionality provided by the Packet Utilization Standard's On-board Queue Model.

### 3.2.3. Frame Store Service

The Frame Store Service provides location awareness between robots by enabling inter-Agent exchange of kinematic and coordinate frame information. The Service provides a data store for relevant objects and a processing capability for computing spatial relationships between objects. The Service generally provides location awareness between Agents, and is generically implemented as a classic tree of coordinate frames and tree-walking routines for calculating coordinate transforms. The Service data stores is generally updated from Agent telemetry.

The Frame Store Service uses the MO Configuration Service [10] to manage an Agent's kinematic and coordinate frame information, including the definition of the location of an Agent's linkages and joints, symbolic naming of coordinate frames, and linkages in coordinate frame trees.

The Frame Store Service also provides a compute service for calculating relative offsets between coordinate frames within an Agent as well as for calculating relative offsets between coordinate frames between Agents.

### 3.2.4. Asynchronous File Transfer Service

The Asynchronous File Transfer Service provides a robust file delivery mechanism between Agents. The Service additionally supports file-based operations in support of tactical and strategic telerobotic operations, including file abstract processing, file directory service and metadata processing, and file-related quality of service specifications for compression and latency.

The Asynchronous File Transfer Service makes use of the CCSDS File Delivery Protocol [11] for basic file transfer operations.

There had been a note relating the MO Action Service to the Asynchronous File Transfer Service, but did not contain any details. How might the MO Action Service play a role in the Asynchronous File Transfer Service?

[NB: Any requests for improvements to CFDP for MOIMS-TEL should be directed to Scott Burleigh; CFDP is currently undergoing its standard CCSDS 5-year protocol review. - DSM]

### 3.2.5. Access Control Service

The Access Control Service authenticates Agents, authorizes their participation in the

Agent network, and ensures the security and reliability of data in transit. The Service ensures that Operators have the proper security credentials to access the Agents under control and that communications between Agents are free from unauthorized modification in transit. The Access Control Service is a fundamental service for telerobotic operations and is a precondition for nearly all other services described in this document.

The Access Control Service will depend upon other CCSDS information security protocols, including:

- Space Data Link Security Protocol [12]
- Security Architecture for Space Data Systems [13]

### 3.2.6. Transfer of Control Service

The Transfer of Control Service mediates requests by Operators to control Agents by ensuring that requests to transfer control between Agents are handled according to an established policy. Transfer of control policies will likely differ according the relative locations of Agents and Operators (local versus remote) and Agent operating mode (test versus operations).

The Transfer of Control Service uses the MO Configuration Service [10] to establish the policies under which control may be transferred as well as to indicate the present state of control, i.e., which Agents are controlled by which Operators.

TODO: Reference to security “plug-in” as is done in the MO specification.

TODO: Distinguish security from command and control policy: Security is out of scope, while command and control policy is a necessary part of the TEL standard, but is yet undefined.

### 3.2.7. Task Decomposition Service

The Task Decomposition Service provides traceability from high-level tasks, down through their decomposition into lower-level components — such as operations and actions — to the telemetry produced by those actions. The Task Decomposition Service helps structure the relationship between action and reaction in a complex telerobotic operations system. Much like the Frame Store Service, the Task Decomposition Service can be thought of as a tree-oriented structure with high-level tasks as the root nodes and low-level commands and telemetry as the leaf nodes. The Task Decomposition Service contains registries of specific task decompositions, which can then be associated with specific instances of the execution of those tasks. By referring to the registry, systems can reassemble the structure of an executed task from the telemetry provided by the Agent for the purpose of presenting the observed behavior of the Agent in a meaningful context. The Task Decomposition Service will be especially useful when reconstructing behavior taking place over varying time-delay, which results in a complex series of overlapping command and telemetry messages.

### 3.2.8. Imaging Service

The Imaging Service is used to capture images from Agent-mounted cameras, and provides information about Image-related products, such as meta-data, data, resolution, planes, etc.

The Imaging Service relies upon the Command Service to command the acquisition of images, and determine the current state of Agent-mounted imaging devices. In addition to the image data itself, the Imaging Service provides camera and image meta-data, such as field of view, image dimensionality and other data describing the acquired image.

### 3.2.9. Video Service

The Video Service provides moving images to the Operator as a natural way of representing the dynamic state of the environment in which the Agent is operating. Telerobotic operations naturally involve the motion of robotic agents and manipulation of their environment, and moving images, or video, is a natural way to represent the state of the environment to an operator. Issues of time delay and communication disruption make the presentation of motion imagery [6] challenging in terms of filtering out irrelevant data (old, poor quality, etc.) from useful imagery.

A video service also provides a method for transmitting non-real-time video and cataloging video for eventual transmission as part of scientific or diagnostic activities. Video service shares some characteristics of a still imagery service in that the imagery is provided by a specific device or sensor that is often mounted on an articulating platform that can be controlled as part of the telerobotic operations system. Video and still imagery data share quality of service attributes that affect the value of the imagery for tactical, strategic and scientific uses (compression, resolution, drop-out, etc.)

The Video Service makes use of the Manipulation Service to control articulating platforms.

The Video Service makes use of the Imaging Service in areas where functionality overlaps, such as in device selection, resolution, optical zoom, etc.

The Video Service uses the MO Configuration Service [10] to enable the following operations:

- **Source Configuration**: Resolution, refresh rate, etc.
- **Source Selection**: Enabling and disabling cameras as active data sources
- **Stream Control**: Quality of service, minimum and maximum allowable compression, minimum and maximum data rates.

#### 3.2.10. Command Service

The Command Service provides a signal that, when sent to an Agent, causes the Agent to perform a function not otherwise covered by the Services described in this document, and for which *some* coordination is required between Agents and/or Operators. The Command Service is analogous to the MO Action Service, but with additional attributes to support the highly interactive nature of telerobotic operations. The Command Service utilizes both the MO Action Service [8] and the MO Configuration Service [10] to provide its functions.

The Commands function provides a list of Commands that are supported by the receiving Agent.

The Command function extends the MO Action service by adding attributes for…

Priority
Priority denotes the insertion point within the Agent's command queue. Choices are Low, Medium, High and Asynchronous. The default is Asynchronous, unless the Agent is using the Sequencer service.
Action and COM Configuration

Name
Name of the Command being sent. Must be one of the Commands included in the receiving Agent's Commands Configuration Message.
Action and COM Configuration

Unique ID
A unique identifier for this instance of a Command. One possible unique identifier would be a concatenation of the sending Agent name and the timestamp of the sending event.
Action and COM Configuration

Source
An identifier for the source of the command. The value for this attribute is arbitrarily chosen to aid in tracing the execution of commands. Source identifiers might include the subsystem or process that generated the command within the sending Agent.

Action and COM Configuration

Target Subsystem
A unique identifier for the subsystem within the receiving Agent that is to receive the Command.
Action and COM Configuration

Arguments
Name, data type and value of the arguments for this particular command.
Action and COM Configuration

#### 3.2.11. Direct Command Service

The Direct Command Service provides a signal that, when sent to an Agent, causes the Agent to perform a function not otherwise covered by the Services described in this document, and for which *no* coordination is required between Agents and/or Operators. For instance, operation of a unique Agent-mounted instrument — such as a laser sinter — would not be standardized under Telerobotic Operations. Telerobotic Operations describes services beneficial to safely operating collaborative human-robot teams in space, but does not address potentially unique Agent characteristics. The Direct Command Service provides a “pass through” mechanism whereby unique Agent capabilities can be addressed within the context of the Telerobotic Operations service standard without needing to standardize operations of unique functions.

The Direct Command Service inherits from the Command Service many required attributes, such as Unique ID, Source and Target, but many of the remaining attributes become optional, such as Priority, and the values used for the parameters become unchecked, such as the Command Name.

#### 3.2.12. Data Product Service

The Data Product Service provides type-specific access to operationally-relevant data collected by Agents. The service is extensible to cover current and future data product types. Some of the current operationally-relevant data products are:

- Point Clouds: Useful for providing depth data for object recognition for manipulation and terrain maps for navigation route planning. Point Cloud formatted data is also used in the collection of scientifically useful information. Point Cloud data can be generated by a variety of devices, including LIDAR, light-field cameras and stereo cameras.
- TODO: Add more data product types.

### 3.2.13. Discovery Service

The Discovery Service provides a method by which to find Agents and their capabilities as they join and leave the network. The Discovery Service maintains a local store containing information on all Agents who are part of the current network environment. Agents may join and leave the network at will, and they broadcast their presence in support of a discovery mechanism. Individual capabilities may also join and leave the network at will.

For instance, a specific camera may become unresponsive and be marked either as unusable or as no longer part of the capabilities of the Agent that once hosted the camera. Physical addition and removal of devices is a more common occurrence in environments with a mix of human and robotic Agents.

The discovery process also applies to the configuration of the communication network itself. For instance, certain types of message traffic may be partitioned from other types of message traffic to aid in network management. The Discovery Service also supports methods for finding these partitions as they are created and destroyed.

The Discovery Service provides a basic aliveness service that periodically “pings” Agents to determine their availability.

The Discovery Service is similar in functionality to the Directory Service, as described in Mission Operations Common Services [10], and may be replaced in a future issue of this document.

TODO: Determine if the Mission Operations Common Services [10] Directory Service satisfies the requirements of the Discovery Service. [Mehran]

### 3.2.14. Administrative Service

The Administrative Service provides system-level managerial and organizational functions for test and control. The Administrative Service uses the Action Service to execute an Echo function and a Shutdown function. The Administrative Service uses the MO Configuration Service [10] to execute a Message Rate Control function.

The Echo function retransmits the received payload to the originating Agent.

The Shutdown function terminates all connections to the receiving Agent and removes the Agent from the collaborative network.

The Message Rate Control function sets and gets the Agent message publishing rate for streaming telemetry.

### 3.2.15. Location Service

The Location Service establishes the relative or absolute position and velocity of of Agents during surface or free-flight operations. The Location Service uses the MO Configuration Service [10] to maintain a database of symbolic names for Agent positions, as well as to periodically report on the location and velocity of Agents.

The Location Name function defines symbolic references to specific locations and resolves those references when requested.

The Position function reports on the position of an Agent.

The Velocity function reports on the velocity of an Agent.

The Location Service has a “history” component that makes it suitable for storing paths.

The Location Service has some utility in managing keep-out zones for robots and humans working in close proximity. Care needs to be taken to distinguish between the Location and Frame Store services, and between services and “data models” in general.

### 3.2.16. Mobility Service

The Mobility Service controls the motion of navigable rovers over a surface or through space (for free-flyers). The Service provides an interface to the greatest common mobility factor for the broadest set of navigable Agents, including wheeled rovers with a variety of mobility and steering mechanism, and legged robots. The Mobility Service provides both Move and Stop functions through its use of the Command Service.

The Move function allows for one of several different types of moves, including simple straight-line moves to complex moves with rotations and translations. Moves can also be associated with tolerances when addressing Agents that are capable of assessing their own performance.

The Stop function stops all motion of the Agent, including Joints, Wheels, etc.

TODO: Either merge Mobility and Manipulation or better distinguish between driving wheeled robots and robots that move across a surface with articulated limbs (that may also be manipulators.) [DSM] comment

### 3.2.17. Configuration Service

The TEL Configuration Service provides information for describing the existence and state of various components that affect Agent operations, such as number of manipulators, joints per manipulator, joint limits, etc. In many ways, telerobots are analogous to highly complex spacecraft. As such, it makes sense to consider how the Mission Operations Service Concept’s Service Configuration specification might support the definition of a Telerobotic Operations Configuration Service. It is recognized that the use of the term “Configuration Service” to refer to both the Mission Operations and Telerobotic Operations concepts is confusing, but the use of an alternate term in the current context would weaken the connection to the robotic use of the term “configuration” to refer the precise physical assemblage as well as the set of robot joint angles used to reach a location in cartesian space. Where the context does not provide sufficient clues as to which Configuration Service is being referred to, we will prepend the term with either MO or TEL to remove the ambiguity.

This section lists specific configuration elements that are of interest during Telerobotic Operations. These elements may also be suitable for inclusion in a Directory Service; robot configuration information is known *a priori*, but changes dynamically during operation (losing a camera, for instance).

The TEL Configuration Service uses both the Action Service and the MO Configuration Service [10] to provide its functions.

The Joint Definition function provides information on the name of joints, frame references, and type of motion (rotary, linear, etc.).

The Joint Configuration function defines the kinematic chain of joints that make up a manipulator.

The Joint State function provides status on an individual joint, such as Enabled, Disabled, Stopped, Stuck, Failed, Overcurrent, Error, etc.

The Joint Sample function provides time-varying data per joint, such as Angle, Velocity, Acceleration, Current, Torque, Temperature, and various Extended State data, etc.

The Dynamic Parameters function provides information on force control modes, controller modes, gains, free-floating base, etc.

## 3.3. Data Model

The Data Model for the Telerobotics set of functions will be detailed in the subsequent Blue Book. One of the challenges will be coordinating and defining the Telerobotics Data Models and the MO SM&C COM data models.

### 3.3.1. Potential Benefits

There is a strong international desire to collaborate on defining Telerobotics standards to reduce the life cycle costs associated with interoperability and cross-agency support in space exploration.

Spaceflight is costly across the development, flight unit production, and launch and operation phases of missions. Spaceflight is also risky to both man and machine. Through collaboration, the international community can contribute to research that will reduce cost and risk. An even greater benefit is when these new technologies increase capabilities or add whole new functions that extend the possibilities of space exploration.

The savings and risk reduction obtained through the development of any component Telerobotic technology is multiplied by the opportunity that interoperability offers us to directly measure and compare similar technologies without a combinatorial increase in development cost. Telerobotic interoperability would allow component technologies to be tested in a rich shared environment: such as an ISS-based test-bed: without the need to create new infrastructure to support each new technology.

## 3.4. Open Questions Regarding Telerobotic Operations

This section lists, in no particular order, some open questions regarding Telerobotic Operations.

### 3.4.1. Real-Time or Near-Real Time Operations

Real-time functions (may be covered already in supervisory functions but here with emphasis on low latency (milliseconds) data exchange)

- Data streaming for direct motion control functions
- Monitoring time critical data (especially for bilateral control)

### 3.4.2. Voice and Audio Streaming

TODO: Determine if a specialized Voice and Audio Streaming service is required for Telerobotic Operations, and if so, describe that service here. [Osvaldo]

- Identify requirements for Voice and Audio Streaming in support of Telerobotic Operations beyond those provided by Voice and Audio Communications [5] and subsequent documents.
- Determine if the Voice and Audio Communications [5] and subsequent documents cover all the uses of voice communication in Telerobotic Operations, including flight/ground, flight/flight, etc.
- Does Telerobotic Operations need its own Audio Streaming service to support Operator voice loop or Operator / Agent voice command?
- Selection of codec and appropriateness of codec
- Relationship between existing Voice and Audio Communications [5] use cases and the case for Telerobotic Operations
- Support for time-delayed and disrupted communication networks

### 3.4.3. Other Relevant CCSDS Standards

There are a number of CCSDS concepts represented in Green books that, while addressing the needs of Telerobotic Operations, are not currently mature enough to be included in this baseline Concept document. Once the following Concepts are developed in more detail, we may revisit their inclusion in future versions of the Telerobotic Operations standard:

- Missions Operations Services Concepts
    - Automation Service
    - Scheduling Service
    - Time Service
    - File Management Service

### 3.4.4. Virtual Telepresence

Do we want Telerobotic Operations to include services related to virtual telepresence, virtual or augmented reality, etc?

# 4. Overview of the Mission Operations Service Framework

## 4.1. Mission Operations Service Introduction

The following descriptions are derived from the CCSDS Mission Operations Services reference documents described in this document. These documents provide a very detailed description of the Mission Operations Services Framework among it's approximately 500 pages. Key elements were extracted and summarized to provide key concepts for the readers.

## 4.2. Mission Operations Services Concept

This CCSDS Mission Operations Services Concept Green Book is an informational report that presents an overview of a concept for a Mission Operations Service Framework for use in spacecraft monitoring and control. It has been prepared by the Spacecraft Monitoring and Control (SM&C) Working Group of the Mission Operations and Information Management Systems (MOIMS) area.

In this context, Mission Operations (MO) refers to end-to-end services between functions, based on the ground or even resident on-board a spacecraft, that are responsible for mission operations.

Standardization of a Mission Operations Service Framework offers a number of potential benefits for the development, deployment and maintenance of mission operations infrastructure:

- Increased interoperability between agencies, at the level of spacecraft, payloads, or ground-segment infrastructure components
- Standardization of infrastructure interfaces, even within agencies, leading to re-use between missions and the ability to establish common multi-mission infrastructure
- Standardization of operational interfaces for spacecraft from different manufacturers
- Reduced cost of mission-specific deployment through the integration of reusable components
- Ability to select the best product for a given task from a range of compatible components
- Greater flexibility in deployment boundaries: functions can be migrated more

easily between ground-segment sites or even from ground to space
- Standardization of a limited number of services rather than a large number of specific inter-component interfaces
- Increased competition in the provision of commercial tools, leading to cost reduction and vendor independence
- Improved long-term maintainability, through system evolution over the mission lifetime through both component and infrastructure replacement

The subjects covered in this volume include:

- Approach to Service Identification
- Service Structure
- Mission Operations Functions
- Identified Mission Operations Services
- Mission Operations Service Concept
- Reference Model
- Message Abstraction Layer [1]
- Common Object model
- Service Specifications
- Language API

### 4.3. Mission Operations Services Message Abstraction Layer

The Mission Operations Services Message Abstraction Layer Recommended Standard [1] defines the Mission Operations (MO) Message Abstraction Layer (MAL) in conformance with the service framework specified in reference, Mission Operations Services Concept.

The MO MAL is a framework that provides generic service patterns to the Mission Operation services defined in reference. These Mission Operations services are defined in terms of the MAL.

The services patterns described in this volume include:

- The *Send* pattern is the basic interaction of which all other patterns can be considered extensions. It is the simple passing of a message from a consumer to a provider. Because there is no message “conversation” implied with a simple *Send*, there is no requirement for a transaction identifier in the message. However, one may be specified. No return message is sent from the provider to the consumer, so the consumer has no indication the provider has received the message. The *Send* pattern is expected to be used for non-critical messages where the possible loss of one or more of these messages is not considered critical. An example would be regular heartbeat-type messages.

- The *Submit* pattern extends the *Send* pattern by providing a return acknowledgment message from the provider back to the consumer. The service specification details the meaning of the acknowledgment message for a specific operation. The *Submit* pattern is used for simple operations that complete quickly but must be confirmed to the consumer.

- The *Request* pattern provides a simple request/response message exchange. Unlike the *Submit* pattern, no acknowledgement is sent upon reception of the request; however, a data response is sent. The lack of an acknowledgement and only the return data response for this pattern means that it is primarily expected to be used for situations where the operation takes minimal time. It is expected that the *Request* pattern is to be used only for operations that complete in a relatively short period of time. If a more extended or indeterminate period is possible then the more advanced *Invoke* or *Progress* patterns should be specified.

- The *Invoke* pattern extends the *Request* pattern with the addition of a mandatory acknowledgement of the initial message. The *Invoke* pattern is expected to be used when there is a significant or indeterminate amount of time taken to process the request and produce the data response. The provision of the service-defined acknowledgement message allows an operation to return supplementary, status, or summary information about the request before processing continues (for example, an identifier used for querying *Invoke* status using another operation).

- The *Progress* pattern extends the *Invoke* pattern with the addition of a set of mandatory progress messages. The type of progress messages and their number is defined by the service and not by the pattern. The *Progress* pattern is expected to be used when there is a significant or indeterminate amount of time taken to process the request and produce the data response, and where monitoring of the progress of the operation is required or the data response is to be returned in blocks.

- The *Publish-Subscribe* pattern for both the consumer and provider register/deregister has a predefined pattern message structure that allows an implementation of the message broker to manage the mapping of consumers to updates and hides this complexity from provider applications.

## 4.4. Mission Operations Services Common Object Model

The Mission Operations Services Common Object Model Recommended Standard defines the Mission Operations (MO) Common Object Model (COM) in conformance with the service framework specified in annex B of Mission Operations Services Concept (reference.

The MO COM is a generic service template that provides a Common Object Model to the Mission Operation services defined in reference [C1]. These Mission Operations services are defined in terms of the COM and the Message Abstraction Layer (MAL) [1].

The services described in this volume include:

- An *Event* is a specific object representing "something that happens in the system at a given point in time." The event service defines a single publish-subscribe operation that supports the monitoring of events generated by other components. An *Event*, as it is a COM object, is identified by the normal object fields (domain, object type, and object instance identifier) with the addition of a string name. The name provides a more human friendly means to identify the *Event*.

- The *Archive* service provides a basic archiving function for COM objects. It follows the Create Retrieve Update Delete (CRUD) principles and allows simple querying of the *Archive* (more complex queries are supported but the specifications of these are outside this standard). As changes are made during the lifetime of an object, this information is distributed to consumers using the service defined operations; as long as these updates follow the COM standard for object identification they can also be stored in a COM archive. By storing these updates in an *Archive*, any historical replay-retrieval functions can correctly reflect the history of the objects.

- The *Activity Tracking* service, or activity service for short, provides the ability to track the progress of activities; an activity is anything that has a measurable period of time (a command, a remote procedure, a schedule etc). The basic service provides the ability to track the progress of MAL operations, but it is expected to be used for other processes where appropriate. It defines an event pattern that supports the reporting of the progress of activities from the initial consumer request, tracking its progress across a transport link, to reception by the provider and execution in that provider. The service uses the event service to report the progress of activities which supports the concept of external monitoring where one component is able to monitor the activities in the system without requiring knowledge of what components are active. This permits the implementation of a single component for the monitoring of activity in the system and also for the archiving of this activity. It also supports monitoring of activities that are passed via a chain of components to a provider; these intermediate components are referred to as relays in this document.

### 4.5. Mission Operations Services Common Services

The Mission Operations Common Services [10] Recommended Standard defines Mission Operations (MO) Common Services in conformance with the service framework specified in Mission Operations Services Concept [3].

The Common Services are a set of services that provide support facilities to the Mission Operation services. These services are defined in terms of the Common Object Model (COM) and the Message Abstraction Layer (MAL).

The services described in this volume include:

- The *Directory* service provides publish and lookup facilities to Service Providers and Service Consumers. It allows providers to publish their location in the form of a URI (Universal Resource Indicator) so that consumers can locate it without having to know in advance the location. Strictly speaking a *Directory* is not required if a well-known service is to be used; however, in most circumstances a *Directory* provides required flexibility in the location of services. The *Directory* is organized in a hierarchical tree structure of directory Nodes. Each Node is identified by a Domain, Network Zone and Session and contains a list of services that are currently available at that node. Alternate networks and sessions are supported by having the same Domain identifier but using different network or session identifiers, for example, for a specific domain there may be several sessions in the *Directory* service and each of those may have a different set of Service Providers.

- The *Login* service allows an operator to provide authentication information to the system. It takes the operator’s credentials and uses a deployment-specific mechanism to authenticate the operator; the result of this is used by the MAL during access control. The *Login* service and the access control provided by the MAL are fully dependent on a deployment-specific security architecture (for example the authentication protocol Kerberos). Both layers (Common and MAL) provide access to, and use of, this security service. Neither implement this themselves. See reference [1] for more information regarding access control.

- When a component of the system requires some input from an operator that is not as a result of an action of the operator, for example, confirmation of a critical automated action, the *Interact* service allows a component to request that information from an appropriate operator.

  Four types of operator interaction are supported:

  - The *Acknowledge* interaction is used when the Operator is required to acknowledge an operation (for example, a simple “Ok”-style interaction).
  - The *Confirm* interaction is used with the Operator is required to confirm or decline an operation (for example, a “Yes/No”-style interaction).
  - The *Choice* interaction is used when the Operator is presented with a list of options to choose from (for example, a drop-down list of choices).
  - The *Get values* interaction is used when the Operator is required to provide a set of values (for example, value entry text fields).

- The *Configuration* service provides the ability to transfer the configuration of a service to and from a Service Provider. It provides facilities for the management of configurations held by a provider if applicable to that implementation. Implementations of the *Configuration* service may support bespoke configuration upload mechanisms (such as file upload). This is supported by the *Configuration* specification where consumers would be notified of a new configuration; however, the details of these upload methods are outside of the scope of the Recommended Standard.

- A historical *Replay* is considered to be a session distinct from other active sessions such as the current live session or simulation sessions. Each *Replay* session is created based on an existing live or simulation session (limited to a specific domain and zone); the *Replay* session is a read-only copy of the base session that can be “browsed” under consumer control. The *Replay* service allows a consumer to create, control, and delete replay sessions. Once a *Replay* has been created, a consumer can:

  - Single-step through history forwards and backwards
  - Play the history forwards and backwards
  - Adjust the replay rate of the replay session
  - Delete the replay session

## 4.6. Mission Operations Services Monitor and Control

The Mission Operations Monitor and Control Services Recommended Standard defines Mission Operations (MO) Monitor and Control (M&C) services in conformance with the service framework specified in the Mission Operations Services Concept Informational Report.

The M&C services are a set of services that enables a mission to perform basic monitoring and control of a remote entity. These services are defined in terms of the Mission Operations Common Object Model (COM) Draft Recommended Standard and the Mission Operations Message Abstraction Layer (MAL) [1].

The services described in this volume include:

- The *Action* service enables consumers to control the remote system, typically a spacecraft; however, there is no restriction on what the remote system may be. *Action* invocation operations include issuing of Action directives by an authorized client to the remote Service Provider, and the subsequent monitoring of the evolving execution status of that Action by both the initiator and other client functions.

- *Parameter* status monitoring is performed by publishing the status of a set of predefined monitoring parameters that contain status information. Monitoring parameters have an evolving status represented by a chronological sequence of status updates over an unbounded lifetime. Status updates may be periodic, change-based, or a mixture of the two. Monitoring parameters are basic types such as strings or integers. Composition of parameters is supported by defining an appropriate aggregation.

- *Alerts* are raised asynchronously to report a significant event or anomaly. Alerts may originate within the remote Service Provider (spacecraft or other controlled system) itself or within an associated ground-based component in response to a transition in some monitored status.Alerts are characterised by an identifier and a set of arguments. However, it is possible that some systems require text-only alert messages where the body of the alert is a free-form text message. For these systems it is possible to define a single alert definition that contains a single string argument; however, it should be noted that translation software shall be required when moving argument-based messages and string-based messages between the two formats.

- The *Check* service allows the consumer to define a set of checks to be applied to parameters and then periodically sample the values of those parameters and check them. If a check is violated (for example, goes outside of the specified limits), the consumer is notified by the generation of a COM event.In addition to this, the Service Provider maintains a list of parameters currently violating any associated checks. Each check has an associated severity level. A minimal list of check types is supported; however, service implementations may support additional custom types. The following is the list of checks that must be supported:

  - A *Limit* checks that the value lies within a specified range
  - An *Expected value* checks that the value is checked against a specified value or value of another parameter
  - A *Delta* checks that the change in value is checked against a pair of thresholds

- The *Statistics* service allows the consumer to associate parameters to defined statistical evaluations (e.g., *min*, *max*, *mean*, *standard deviation*) and periodically sample the values of those parameters and evaluate them. The resultant statistics evaluations are provided to consumers who register interest in them. The service allows the set of parameters associated with a statistic to be added to and deleted from; however, it does not allow new statistic algorithms not currently supported by a Service Provider to be defined. For example, if a Service Provider supports *min* and *max* statistics, it is possible to add parameters to these statistical functions, but it is not possible to add a new statistic "standard deviation," as this is not supported by the Service Provider.

- Another logical extension to basic monitoring is data aggregation. This *Aggregation* service provides the capability to acquire several parameter values in a single request. The aggregation might be one of the following:

  - Predefined by the Service Provider, e.g., housekeeping parameters;
  - Predefined at runtime by the consumer, e.g., a diagnostic report.

  For example, the user may request a data product that comprises the accelerations and angular rates of the spacecraft. This would be acquired by reading the appropriate gyros and accelerometers onboard and returning the data as a set. However, if the Service Provider is actually on the ground, for example an MCS kernel acting as a proxy, then it may collate the set from its current "state vector" of all reported parameters.

- A functional extension of the other services is to add the engineering unit conversion capability. For example, using this *Conversion* service with the parameter service, when consumers register for updates for a specified parameter, they receive a value that represents the parameter being read, for example, a quantized temperature value with a unit rather than the raw data value.

# 5. Document Roadmap

Description of approach to developing the subsequent Blue Book.

## 5.1. Testing and Prototyping

In this section, we describe the specific methods that participating Agencies will use to look for opportunities to perform early tests of Blue Book prototypes. See Concept Paper, Section 2.7, “Testing and Prototyping.”

The Telerobotic Operations Working Group encourages participating organizations to look for opportunities to involve their technology development laboratories in Agency field-testing as a way to provide opportunities for interoperability testing. Although we do not anticipate directly supporting the development of a common communications network infrastructure between participating Agencies, we will encourage the development of such infrastructure as a way of reducing the costs associated with the independent prototyping and test activities required of CCSDS standards development groups. The members of the Telerobotic Operations Working Group will actively promote the early prototyping and field-testing of the messages, APIs and services described in the subsequent Blue Book.

To encourage the widest possible test coverage, prototypes will be encouraged in different computer languages, such as C++ and Java, and testing will occur against both actual and virtual robotic assets. We expect each official prototyping effort to implement all defined messages and APIs; however, due to the effort required, individual prototypes may implement simplified versions of the specified services. When taken together, all services will receive at least one complete implementation within the full set of Blue Book prototypes.

## 5.2. Eliminating Overlap with Other Standards

Describe approach to ensuring that Telerobotics is not defining competing or duplicative standards. Outreach and surveillance.

- There is a new focus on identifying and eliminating overlap between standards
- Telerobotics has committed to working with other WGs to eliminate duplication of effort
- Need to identify local “experts” with knowledge of other CCSDS areas and of the Telerobotics domain
- Corollary: Where TEL exists in the overall system of CCSDS standards. See Figure 2, A Strawman Gap Analysis.
- Work with the File Transfer BoF (new as of Spring 2014) to determine overlap between the file transfer needs of Telerobotic Operations and the generic file transfer and management functions envisioned for the new BoF.

## 5.3. Selection of Reference Implementation Language

In an effort to continuously test and improve upon the emerging Telerobotic Operations standard, the Telerobotic Operations Working Group will select a language in which to create implementation of the specification to be used as a definitive interpretation for the specification. This Reference Implementation will be developed concurrently with the specification and test suite so as to verify that the specification is implementable and testable. The Reference Implementation will be the standard against which other implementations will be measured and will help to clarify intent of specification where other sources of information are lacking.

It is anticipated that the selected reference implementation language will be either Java or C++.

# 6. Other Industry Standards

There exist a number of relevant robotics standards that should inform the work of the CCSDS Telerobotic Operations Working Group during its development of its subsequent Blue Book. We include short descriptions of these standards in this Section; refer to the referenced standard itself for a complete description.

## 6.1. Joint Architecture for Unmanned Systems

Originally developed by the United States Department of Defense in 1998, JAUS is an open architecture for the domain of unmanned systems, featuring vehicle platform independence, mission isolation, computer hardware independence, technology independence, and operator use independence.

The JAUS standards are owned and developed by the Society of Automotive Engineers (SAE) under the Aerospace Standards Unmanned Systems Steering Committee (AS-4). All of the standard documents that define JAUS can be purchased online directly from SAE.

JAUS includes several standards specification documents, including network transport, interface definition language, core services, mobility services, human machine interface services and manipulation services. JAUS uses a Service Oriented Architecture (SOA), and is therefore fundamentally compatible with the SOA-based CCSDS Mission Operations services. The JAUS standard focuses on the operation of autonomous ground, air and undersea vehicles, but does not address space-based operations or human-robot collaboration.

Several implementations of the JAUS standard exist, including OpenJAUS, The JAUS Toolset (JTS), and JAUS++. OpenJAUS is a commercial product for non-academic uses, while JTS is a JAUS-compliant system design tool. JAUS++ is an Open Source implementation of many of the JAUS services in C++. There do not seem to be any implementations of the JAUS standard in Java or other languages, although JTS can emit Java code as part of its code-generation process.

The JAUS service set is well-aligned with the service set being defined by the CCSDS Telerobotic Operations Working Group, which has reviewed the service set specification and incorporated lessons from the JAUS service set architecture into the Telerobotic Operations service set. Once work begins on the Blue Book, the Telerobotic Operations Working Group will perform a similar analysis of the JAUS message set to ensure that the Telerobotic Operations message set is not necessarily incompatible at a conceptual level.

## 6.2. Robot Application Programming Interface Delegate

The Robot Application Programming Interface Delegate (RAPID) is a set of software data structures and routines that simplify the process of communicating between multiple diverse robots and their command and control systems. RAPID is not intended to be an all-encompassing API for robot communication, but rather it's a compatibility layer that permits tools and robotic assets to exchange data and information and allows operators to communicate with heterogeneous robots in a uniform way. RAPID is a compatibility layer that delegates information between robots that speak different languages.

The RAPID specification includes definitions and APIs for messages and services that support collaborative, supervisory telerobotic operations over near-Earth time delay. RAPID is not a middleware specification, although safety and time-delay capabilities do imply requirements on implementing middleware systems. As currently implemented, the RAPID system can be considered a software reference implementation for remote operations.

RAPID has been in development and use within NASA since 2007, and current development and test is supported by the Space Technology Mission Directorate through the Game Changing Development Program's Human-Robotic Systems Project and Human Exploration Telerobotics Project.

Members of the RAPID project are contributing to the work of the Telerobotic Operations Working Group.

## 6.3. Coupled-Layer Architecture for Robotic Autonomy

TODO: Write CLARAty description.

## 6.4. Sensor Model Language

The Open Geospatial Consortium's Sensor Model Language (SensorML) [14] "provides a robust and semantically-tied means of defining processes and processing components associated with the measurement and post-measurement transformation of observations. This includes sensors and actuators as well as computational processes applied pre- and post- measurement."

Like CCSDS, the Open Geospatial Consortium is an international standards organization with membership from the space community. The SensorML standard is an outgrowth of OGC's Sensor Web Enablement (SWE) activity, which sought to "enable a "Sensor Web" through which applications and services will be able to access sensors of all types, and observations generated by them, over the Web."

The SensorML effort should inform the Telerobotic Operations standard definition effort in the way TEL enables the exchange of operations-enabling science data products via the Data Product Service.

### 6.5. Robot Operating System

The Robot Operating System (ROS)…

TODO: Finish describing the Robot Operating System. [David]

# Annex A

*Lexicon*

This Lexicon contains definitions of select terms and instructions on how to render terms contained in this Green Book into other languages and/or ontologies. This Lexicon is intended to document decisions related to the selection of the normative terminology used by the Telerobotic Operations Working Group. This Lexicon will hopefully eliminate use of alternative textual references in the main body of the Book and serve as a historical record for lexical decisions made during the editorial process.

Where possible, link terms in the body of the Green Book to entries in the Lexicon using Google Docs' Bookmark feature. First, create a Bookmark for the Lexicon entry. Once the Bookmark is created, Link the terms in the body to the Bookmark in the Lexicon.

*Only create Bookmarks for entries that are referred to from the body text; at some point in the future, unused Lexicon entries may be removed, and the presence of a Bookmark for a Lexicon entry indicates that the entry is in use.*

For Lexicon entries with source definitions from other documents, create a link to the other document from the Lexicon term. See Action for an example.

Create an entry in the Definitions, Nomenclature and Conventions section for the most important terms from this section that are referenced from the body of the document where the definitions of those terms are drawn from other CCSDS documents referenced herein.

Additional Lexicons or Annexes may be created to offer more in-depth comparisons between participating organizations' approaches to collaboration, or as corporate knowledge bases for other significant discussions and/or findings made by the Telerobotic Operations Working Group during Green Book deliberations. Definitions from the SANA Registry of CCSDS Terms will be marked "(CCSDS Term)." Definitions common to the Telerobotic Operations field will be marked "(Term of Art)."

**Action**: An atomic (non-interruptible) control directive of mission operations (equivalent to a telecommand or ground-segment directive) that can be initiated by manual or automatic control sources, via the M&C service. An action may have arguments and its evolving status can be observed. An action can typically apply pre- and post-execution checks. (CCSDS Term)

**Activity**: An automated mission operations function; typically an operations procedure, batch task or other software process. An activity can be individually scheduled or initiated. In principle, an activity is non-atomic, has duration, and can be controlled once initiated. An activity may have arguments and its evolving status can be observed. An activity may generate multiple actions, and its behaviour can be dependent on status observed at runtime. (CCSDS Term)

**Adapter**: In a Service Oriented Architecture (SOA) context, a software component that implements a higher-level service in terms of a lower-level service or specific technology. In this way different adapters can map a high-level service onto different underlying technologies, transparently to all higher layers including the application. Adapters can also wrap non-service-oriented applications so that they can be used as Service Providers in SOA. (CCSDS Term)

**Agent**: An "Agent" is an entity that can be monitored or commanded Telerobotic Operations services. When referring to a robot or class of robots, the term "Robot" can be used in place of "Agent." When referring to a human or class of humans, the term "Astronaut" or "Operator" can be used in place of "Agent," depending on the situation. (Term of Art)

**Alert**: An asynchronous notification, such as a non-nominal event, of significance to mission operations. Alerts may be used to notify such events to operators, initiate an automatic response, or synchronize asynchronous functions. Alerts may have arguments. (CCSDS Term)

**Application Programmers' Interface**: The definition of the exposed or "public" interface to a software component that can be used by another software component. In a Service Oriented Architecture context, an API corresponds to a language- or technology-specific implementation of an abstract service specification. This constitutes the code classes, types and functions utilised by a programmer when implementing the Service Provider and Service Consumer. (CCSDS Term)

**Argument**: A run-time parameter provided to various control items on invocation, e.g., telecommand arguments. Arguments apply to actions, activities and alerts among other items. (CCSDS Term)

**Bilateral Control**: A system of teleoperation control in which control signals flow from the operator to the robotic agent to carry out a task, while at the same time, feedback signals flow from the robotic agent to the operator, usually to provide an indication of the forces experienced by the robotic agent as a method to improve the operability of the system. *(Term of Art)*

**Capability**: A capability is a core ability of a system, or a function offered by a service.

**Capability Set**: A grouping of functions, offered by a service, that are logically related. Capability sets are used to decompose a service into smaller functional areas. (CCSDS Term)

[DSM] Consider defining a Minimum Capability Set, which would be the minimum set of supported services that qualifies an Agent for adherence to the standard. Consider different Minimum Capability Sets for different classes of Agents: navigable rovers might be required to provide the Location service, for instance. Required elements will tend to support the “mission statement” of the standard: safe, supervisory and collaborative telerobotics over time-delayed and disruption prone networks.

**Common Object Model**: The generic service template that Mission Operations Services are defined in terms of.

**Consumed Service Interface**: The API presented to the consumer component that maps from the Service operations to one or more Service Data Units (SDUs) contained in MAL messages that are transported to the provided service interface.

**Domain**: A namespace that partitions separately addressable entities (e.g., actions, parameters, alerts) in the space system. The space system is decomposed into a hierarchy of domains within which entity identifiers are unique.

**Dynamic Object/Dynamically Instantiated Object**: An entity that is instantiated, invoked or created at runtime based on a static definition. Examples include actions, alerts and activities. Multiple copies (instances) of such objects may exist concurrently, but all share a single definition.

**Encapsulation**: A software design approach that provides code users with a well-defined interface to a set of functions in a way which hides their internal workings or means of implementation. In object-oriented programming, the technique of keeping together both data structures and the methods (procedures) which act on them.

**Event**: A time-stamped message, containing (changes in) information about information objects associated with a service, that is exchanged across service interfaces and potentially stored in service history.

**Exposed Interface**: A published (or "public") interface, provided by a software component, that is available for use by other software components.

**Hardware Component**: A complex physical entity (such as a spacecraft, a tracking system, or a control system) or an individual physical entity of a system (such as an instrument, a computer, or a piece of communications equipment). A hardware component may be composed from other hardware components. Each hardware component may host one or more software components. Each hardware component has one or more ports where connections to other hardware components are made. Any given port on the hardware component may expose one or more service interfaces.

**Management Service Interface**: A service interface that exposes management functions of a service function contained in a component for use by Service Consumers.

**Mission Operations Services**: A suite of end-to-end application-level services that constitute a Service Oriented Architecture for space mission operations.

**Object/Information object/service object**: Information objects are passed across a service interface. These are defined in the information model of the service.

**Object Instance**: Alternative term for object that distinguishes between the multiple run-time invocations of an object and their associated static definition (see statically and dynamically instantiated objects).

**Operation**: In object-oriented programming, a method, function, or message defined for a class of objects. In the mission operations services context, a control primitive that can be performed across the service interface.

**Parameter**: An item of mission operations status information that can be individually subscribed to by a service consumer, via the M&C service. A parameter has multiple attributes, including: raw value, engineering value, validity, check status and (optionally) statistics.

**Plug-in**: A software component that can be integrated with other components conforming to the same Service Oriented Architecture, without the need to modify the implementation of other components. In the MO context, this could apply to both Service Consumer / provider applications and infrastructure components that implement lower levels (protocol layers) of the service interface.

**Protocol Data Unit**: Elemental data message for exchange between peer service

layers of two applications using a particular implementation protocol.
provider,

**Provided Service Interface**: A service interface that exposes the service function contained in a component for use by Service Consumers. It receives the MAL messages from a consumed service interface and maps them into Application Program Interface (API) calls on the provider component.

**Proxy**: In the context of MO, a proxy function or component is one that acts locally in the place of a remote Service Provider, such as a spacecraft. There is a proxy function for each service. It provides a dual role. Firstly it provides a permanent point of contact for service consumers where the link to the remote Service Provider is intermittent, maintaining an image of current status, buffering operations and managing the service history. Secondly it can act as an isolation layer and adapter to actual protocols employed on the space-ground interface.

**Replay**: The act or interface associated with viewing data from a service history in the same manner as live operation. Service events are dynamically replayed over an evolving time period.

**Retrieval**: The act or interface associated with of withdrawing a data set by a time range from a service history. Retrieval is mainly intended for fast access to a block of service history for display of data trends or logs over a period of time, or to be used in analytical tasks.

**Service**: A set of capabilities that a component provides to another component via an interface. A Service is defined in terms of the set of operations that can be invoked and performed through the Service Interface. Service specifications define the capabilities, behaviour and external interfaces, but do not define the implementation. (CCSDS Term)

**Service Capability Set**: The specification of Services is based on the expectation that different deployments require different levels of complexity and functionality from a service. To this end a given service may be implementable at one of several distinct levels, corresponding to the inclusion of one or more capability sets. The capability sets define a grouping of the service operations that remains sensible and coherent; it also provides a Service Provider with an ability to communicate to a consumer its capability. [DM] Reconcile this with the definition of Capability Set.

**Service Configuration Data**: Configuration data (in the form of a database or other file) that defines the characteristics of a specific instance of a service. Typically this identifies the information objects that exist in the context of a particular service, for a particular domain, e.g., the specific actions, parameters and alerts applicable to a given spacecraft would be defined in the M&C service configuration data for that spacecraft. Access to the service configuration data is required by both Service Consumer and Service Provider (or its proxy).

**Service Consumer**: A component that consumes or uses a service provided by another component. A component may be a provider of some services and a consumer of others.

**Service Data Unit (SDU)**: A unit of data that is sent by a service interface and is transmitted, semantically unchanged, to a peer service interface.

**Service Directory**: A service directory is an entity that provides publish and lookup facilities to Service Providers and consumers.

**Service Extension**: Addition of capabilities to a base service. A service may extend the capabilities of another service with additional operations. An extended service is indistinguishable from the base service to consumers such that consumers of the base service can also be consumers of the extended service without modification.

**Service History**: The operational data archive for a service. This is the data required to reconstitute a historical view of information at the service interface, either using replay or retrieval access methods. It corresponds to the persistent sequence of all service events over a period of time, to which a Service Consumer could have subscribed. Examples of service histories include parameter history, action history and alert history. Alternative implementations are possible, based on archiving of protocol messages (e.g., packets) and re-processing.

**Service Instance**: A deployment copy of a service, typically for a specific domain. A Service Provider constructs and publishes a service instance. Service Consumers may then subscribe to that service instance.

**Service Interface**: A set of interactions provided by a component for participation with another component for some purpose, along with constraints on how they can occur. A service interface is an external interface of a service where the behaviour of the Service Provider component is exposed. Each service will have one defined “provided service interface.” and may have one or more “consumed service interface” and one “management service interface.” (CCSDS Term)

**Service Oriented Architecture**: Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It is not itself a solution to domain problems but rather an organizing and delivery paradigm that enables one to get more value from use both of capabilities which are locally "owned" and those under the control of others. [15]

**Service Provider**: An application or component that offers a service to another by means of exposing a Service Interface, while hiding details of its implementation.

**Session**: A session defines the time-frame for a service. A session may be live or historical, real or simulated. A Service Consumer may join any existing session by subscribing to a service for that session. Within a given system there may be multiple concurrent sessions, to support simulated and/or historical replay sessions in parallel with live operations. Within service history there may be multiple session histories, corresponding to live operations and simulated sessions.

**Software Component**: A software unit containing the business function. Components offer their function as services, which can either be used internally or which can be made available for use outside the component through provided service interfaces. Components may also depend on services provided by other components through consumed service interfaces.

**Static Object Statically Instantiated Object**: An entity that is effectively instantiated at operations preparation time, e.g., a parameter. It has a static portion (the definition) and a dynamic portion (its current status). See also dynamically instantiated object.