|  |  |  |
| --- | --- | --- |
| SCISYS-Colour-Horizontal-Medium | UK Space Agency | Image result for ccsds logo |
|  | | |
| FrontCover_BGCCSDS Standardisation Support  CCSDS MAL and SEDS | | |
|  | | |
| SSL/yy-nnn/DOC/001  Issue , 23/02/2017 | | |
| ,  | | | | |

Intentionally Blank

|  |  |
| --- | --- |
| SCISYS-Colour-Horizontal-Medium | |
|  |  |
| Project: | CCSDS Standardisation Support |
| Doc Type: | Technical Note |
| Title: | CCSDS MAL and SEDS |

|  |  |
| --- | --- |
| Document Control Information | |
| **Contract/ ITT Reference** |  |
| **SCISYS Reference** | SSL/yy-nnn/DOC/001 |
| **Issue** |  |
| **Issue Date** |  |
| **Customer Reference** |  |
| **Classification** |  |

|  |  |  |
| --- | --- | --- |
| Role | **Name(s)** | **Signature(s)** |
| **Author** |  |  |
| **Reviewed** |  |  |
| **Authorised for Release** |  |  |

**NOTICE**

The contents of this document are the copyright of SCISYS UK Limited and shall not be copied in whole, in part or otherwise reproduced (whether by photographic, reprographic or any other method) and the contents thereof shall not be divulged to any other person or organisation without the prior written consent of SCISYS UK Limited.

© SCISYS UK Limited 2017

# Distribution

|  |  |
| --- | --- |
| Copy Number(s) | **Recipient** |
|  |  |
|  |  |
|  |  |

# Issue Record

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | **Issue Date** | **Sections Affected** | **Relevant Information** |
| **1.0** |  | All | Not Yet |

# Table of Contents

[Distribution ii](#_Toc476321278)

[Issue Record ii](#_Toc476321279)

[Table of Contents iii](#_Toc476321280)

[Table of Figures iii](#_Toc476321281)

[Table of Tables iii](#_Toc476321282)

[1. Introduction 5](#_Toc476321283)

[1.1 Document References 6](#_Toc476321284)

[1.2 Definitions 7](#_Toc476321285)

[1.2.1 Acronyms 7](#_Toc476321286)

[2. Introduction to SOIS EDS 8](#_Toc476321287)

[2.1 SOIS EDS Datasheet 9](#_Toc476321288)

[2.2 Simple Example of Datasheet Use 11](#_Toc476321289)

[3. Introduction to MO MAL 12](#_Toc476321290)

[4. Analysis 14](#_Toc476321291)

[4.1 Areas of Overlap Between the Standards 14](#_Toc476321292)

[4.2 Specifying Interfaces 16](#_Toc476321293)

[4.3 Detailed Comparative Analysis 17](#_Toc476321294)

[4.4 Mapping between SOIS EDS and MO MAL 19](#_Toc476321295)

[5. Prototyping 20](#_Toc476321296)

[6. Recommendations and Conclusion 21](#_Toc476321297)

# Table of Figures

[Figure 1 Two hypothetical missions using CCSDS Standards 4](#_Toc476321262)

[Figure 2‑1 CCSDS SEDS Concept 7](#_Toc476321263)

[**Figure 2‑2 SOIS Reference Communications Architecture** 7](#_Toc476321264)

[**Figure 2‑3 SOIS EDS Device Datasheet Contents** 8](#_Toc476321265)

[**Figure 2‑4 Parameters and Commands on an interface** 9](#_Toc476321266)

[**Figure 2‑5 Role of SOIS EDS within SOIS-based OBSW** 10](#_Toc476321267)

[Figure 7 CCSDS MO Scope 11](#_Toc476321268)

[Figure 8 Details of a MO Service 12](#_Toc476321269)

[Figure 9 Transformation of MAL into technology-dependent interface specifications 12](#_Toc476321270)

[Figure 10 Overlap Between Standards 13](#_Toc476321271)

[Figure 11 Binary interface to the Device expressed as a CCSDS EDS 13](#_Toc476321272)

[Figure 12 Sequence diagram: adjusting a setting on a device at the request of the end user 14](#_Toc476321273)

[Figure 13 Interface Formalism properties 16](#_Toc476321274)

[Figure 14 XML structure of EDS and MAL 16](#_Toc476321275)

[Figure 15 MAL Interaction Patterns 17](#_Toc476321276)

[Figure 16 EDS Interaction Patterns 17](#_Toc476321277)

# Introduction

Founded in 1982 by the major space agencies of the world, the CCSDS is a multi-national forum for the development of communications and data systems standards for spaceflight, with the goal of enhancing governmental and commercial interoperability and cross-support, while also reducing risk, development time and project costs. Within that organisation, working groups have been tasked with looking at different areas of interoperability, specifically:

* Mission Operations and Information Management Services (MOIMS), covering the interface between operations teams and the spacecraft.
* Spacecraft On-board Interface Services (SOIS), covering the interface between spacecraft and on-board systems or devices.

This division of responsibility can be illustrated by an example whereby a hypothetical university designs, builds and is involved in the operation[[1]](#footnote-1) of a simple on-board instrument.

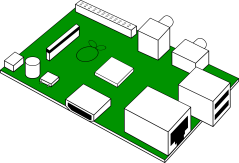
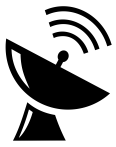
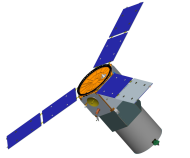
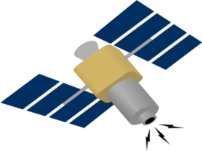
* SOIS is responsible for the information the university supplies to the agency in order to integrate the instrument into the overall on-board platform.
* MOIMS is responsible for the information the university supplies to the agency in order to operate the instrument during the mission lifecycle.

ESA

SPACELINK

SPACELINK

MOIMS



MOIMS

SOIS

SOIS

Specifies,

builds,

operates

ESA

ESA

NASA

Figure 1 Two hypothetical missions using CCSDS Standards

If two copies of similar devices fly on different spacecraft operated by different agencies, then, if the CCSDS standards are applied in all cases, the result is minimal extra work for the university. The same principles apply in more complex cases where the client, designer, manufacturer and operator are not the same, or where there are multiples of each.

This technical note is aimed at examining the CCSDS standards produced for these two areas, with an eye to looking for opportunities to transfer lessons learned between them.

## Document References

| **ID** | **Reference Document** | **Reference** | **Version** | **Date** |
| --- | --- | --- | --- | --- |
| **[SEDS]** | XML Specification for Electronic Data Sheets | 876x0 | 2016 draft | 19/08/2016 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Definitions

### Acronyms

| **Acronym** | **Definition** |
| --- | --- |
| API | Application programming Interface |
| CCSDS | Consultative Committee for Space Data Systems |
| ECSS | European Cooperation for Space Standardization |
| EDS | Electronic Data Sheet |
| MAL | Message Abstraction Layer |
| MO | Mission Operations |
| OBSW | Onboard Software |
| RMI | Remote Method Invocation |
| SOIS | Spacecraft Onboard Interface Services |
| UML | Unoversal Modelling Language |
| XML | eXtensible Markup Language |

# Introduction to SOIS EDS

Electronic Data Sheets (EDS) is a concept that has been proposed to allow the capture of the relevant information about a piece of equipment. This should capture the relevant aspects not just to enable an efficient exchange of information (easing its maintainability, enforcing consistency, etc.), but also enabling the development process of related software to be supported by the use of model-based software engineering techniques.

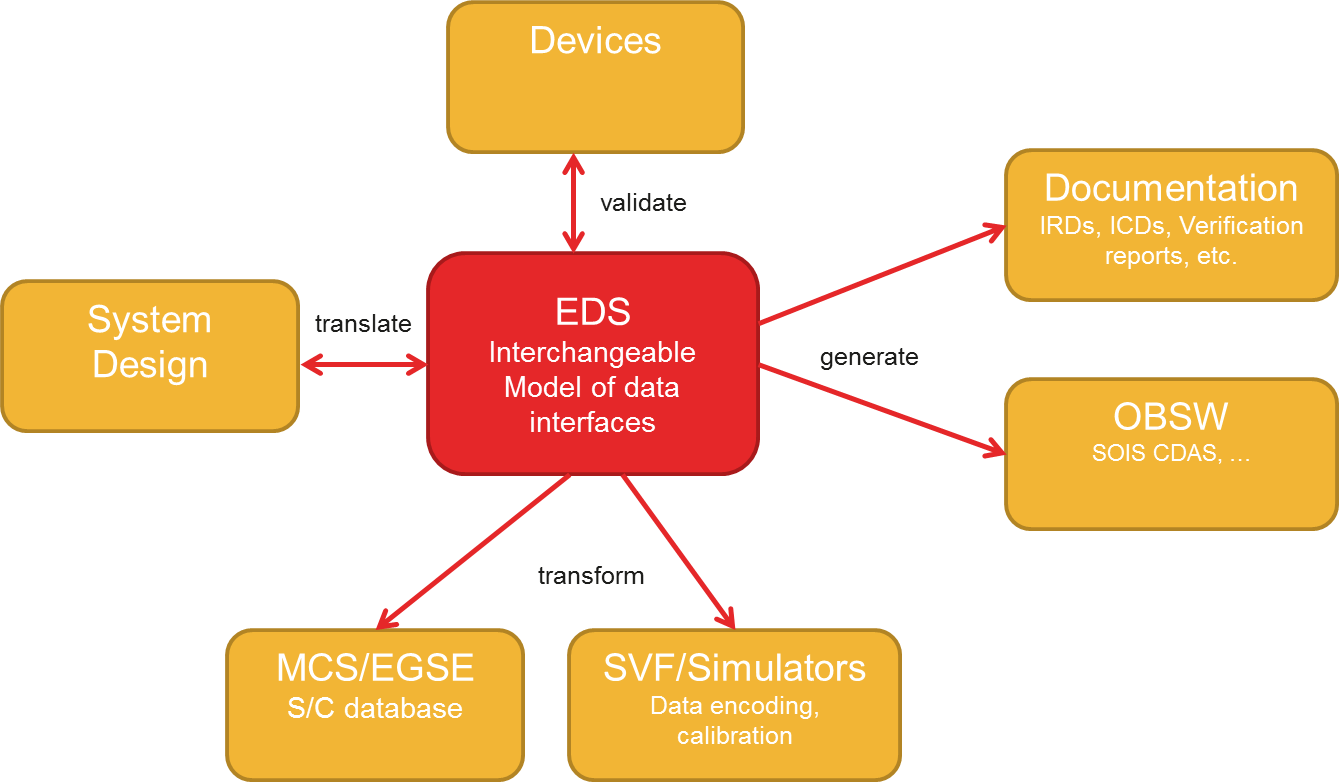


Figure 2‑1 CCSDS EDS Concept

The SOIS standard for Electronic Data Sheets [SEDS] takes the form of an XML schema[[2]](#footnote-2) designed for *tool interchange*, i.e. exchanging device data between two software systems. It forms part of the SOIS Reference Communication Architecture, as shown below.



**Figure 2‑2 SOIS Reference Communications Architecture**

Some portions of the application-level functionality supported by hardware device are sufficiently universally fixed to be usefully specified as an application-level service, i.e.:

* Clocks may support the Time Access Service.
* Mass Memory Devices may support the File and Packet Store Services
* Dynamically reconfigurable devices may support the Device Enumeration Service.

With those exceptions, every aspect of the interfaces and functionality provided by the device can be specified in the EDS for that device. Such a device datasheet defines the interpretation and contents of the messages exchanged by applications and devices across the SOIS Subnetwork Layer, which provides an abstract model of services that can be mapped to the actual subnetwork-specific protocols used. This layer covers sending and receiving discrete packets, accessing remote memory, synchronising with the subnetwork, and also the discovery and test of devices on the subnetwork.

By specifying the device data interface in terms of this model, it becomes possible to determine the correctness and completeness of a device datasheet in isolation from the actual OBSW that will be used to communicate with the device in any particular case. This validated datasheet can then be used as an input to the development and testing of those systems that interact with the device (e.g. the spacecraft OBSW, checkout systems, etc.).

## SOIS EDS Datasheet

A SOIS EDS Datasheet is a package that can contain definitions of:

* Interfaces that allow two-way data interchange, within the scope of a single device.
* Components that map a set of *provided* interfaces to a set of *required* interfaces.



**Figure 2‑3 SOIS EDS Device Datasheet Contents**

By convention, a complete datasheet for a device should contain at least two interfaces:

* The Device-Specific Access Interface; the lowest-level access to all raw decoded data transmitted to and from the device.
* The Device-Specific Functional Interface; higher-level access to calibrated or derived data.

Both of these interfaces are device-specific because different devices support different sets of data. These are split to allow missions the option of supporting either one, or both[[3]](#footnote-3).

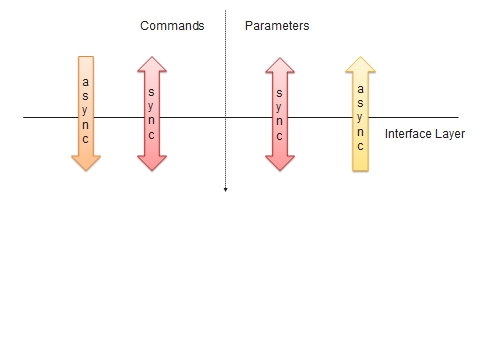
In the typical case, there will be a single component providing each interface, and the component implementing the higher-level interface will be defined in terms of the lower-level one. The lowest-level component will require one or more subnetwork-level interfaces.

Mapping the device-specific interfaces defined in the datasheet to actual APIs or messaging interfaces used by a specific OBSW architecture is explicitly not the concern of a datasheet; otherwise the same device datasheet could not be used when the same hardware device is used for missions of different software architectures.

All interfaces provided and required are explicitly defined within a datasheet; there is no privileged treatment or special-casing for standardised interfaces. The datasheet construct used to define interfaces can be used to specify both high-level functional interfaces (e.g. an API, or messaging interface with regular encoding) and low-level binary interfaces containing arbitrarily-encoded data, as commonly produced by device hardware.

A key characteristic of SEDS interfaces is that while they support 2-way data exchange, they are partitioned into:

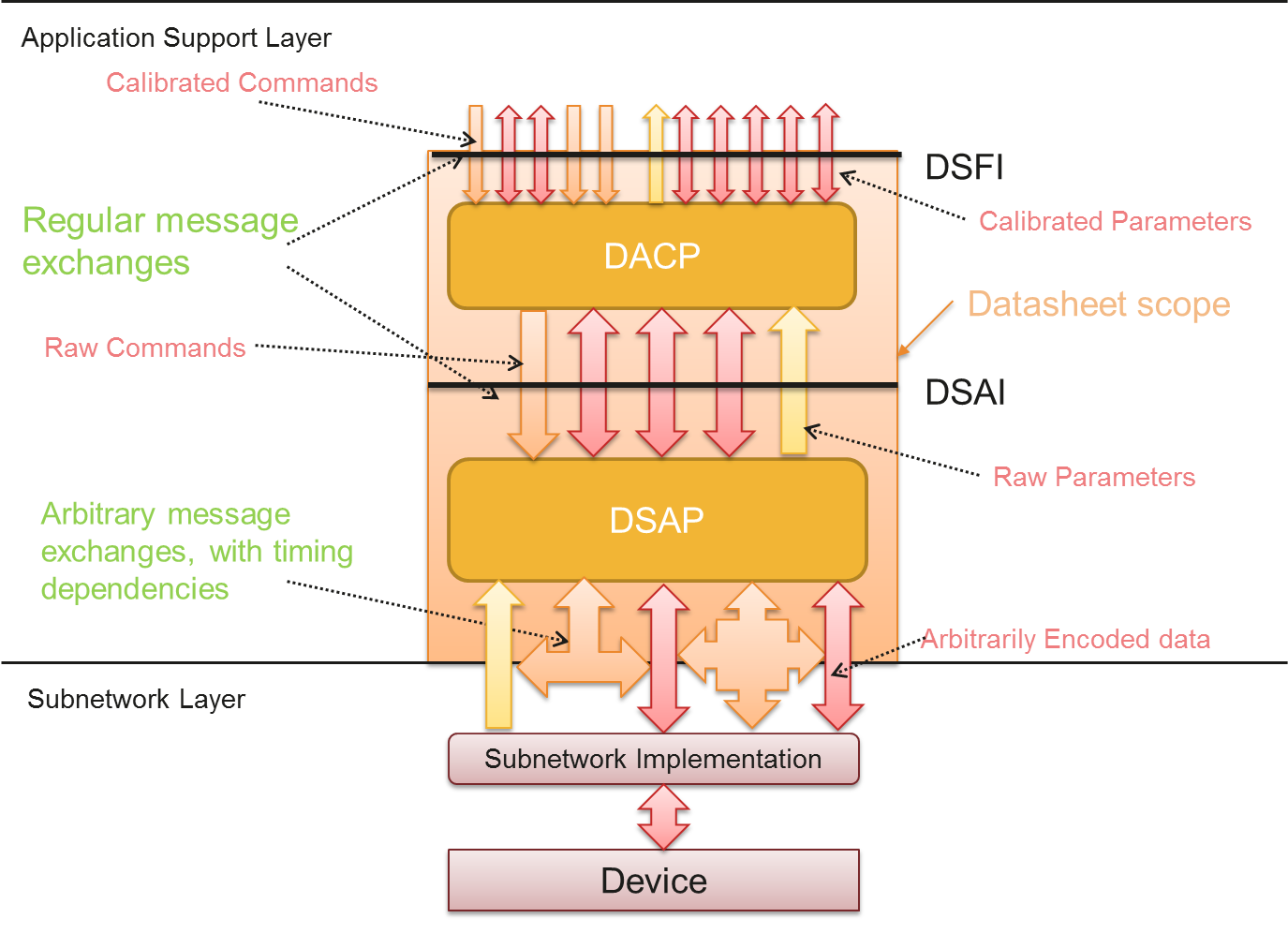
* Parameters**[[4]](#footnote-4)**; messages coming from the device, plus those 2-way exchanges whose sole purpose is to pull information from a device.
* Commands; messages sent to a device, plus 2-way exchanges with any purpose other than reading a single parameter.



**Figure 2‑4 Parameters and Commands on an interface**

As a consequence of the above, SEDS interfaces are able to not only *specify* a new interface (a capability shared by many other similar component systems) but to *capture* an existing interface. This includes cases where that interface was designed and implemented without knowledge of the SEDS or SOIS.

## Simple Example of Datasheet Use



**Figure 2‑5 Role of SOIS EDS within SOIS-based OBSW**

The above diagram shows the information specified by a datasheet being directly used[[5]](#footnote-5) to communicate with a device. In it:

* The Device Specific Access Protocol takes an arbitrary set of encoded binary messages from the subnetwork level and rearranges them into a known and finite set of raw commands and parameters.
* The Device Abstraction Control Procedure further maps those access-level commands and parameters them into calibrated parameters and commands, which have semantically meaningful values in terms of engineering units.
* The Subnetwork Implementation is a thin software layer implementing a standardised subnetwork protocol (e.g. ECSS SpaceWire) referenced, but not specified, in the datasheet.

# Introduction to MO MAL

*Monitor and Control*

*Common Infrastructure*

*Planning*

*Navigation*

*Data Product Distribution*

*Scheduling and Automation*

*Software Management*

*File Transfer and Management*

*Telerobotics*

Figure 7 CCSDS MO Scope

CCSDS is standardising a set of services for Mission Operations. These services define a single specification for the exchange of similar information.

To support these standardised services CCSDS has also defined an open architecture and framework that is:

* Independent from technology
* Able to integrate new and legacy systems of different organisations
* Designed to support the long lifetimes of space missions
* Based on a Service Orientated Architecture (SOA)
* Allows defining new bespoke services for a mission-specific need.

Service

Provider

Consumer

Operation

Figure 8 Details of a MO Service

Each MO service, whether standardised or bespoke, is defined by a set of operations that the provider of the service makes available to be invoked by the service consumer. Each operation is defined from a template specified by an interaction pattern; one of send/submit/request/invoke/progress/pubsub. Each such pattern has a list of the messages that must be specified (e.g. request and response for the request pattern).

The MO concept is supported by the MO framework, which defines the structure of an MO application, provides a generic model for data, supports generic facilities such as archiving, and provides separation from technology

At the core of the framework is the Message Abstraction Layer (MAL), which defines a standard XML notation for service specifications. These abstract specifications then get transformed into the appropriate representation for whatever underlying technology is used at implementation time (e.g. Interface Definition Language for Corba).

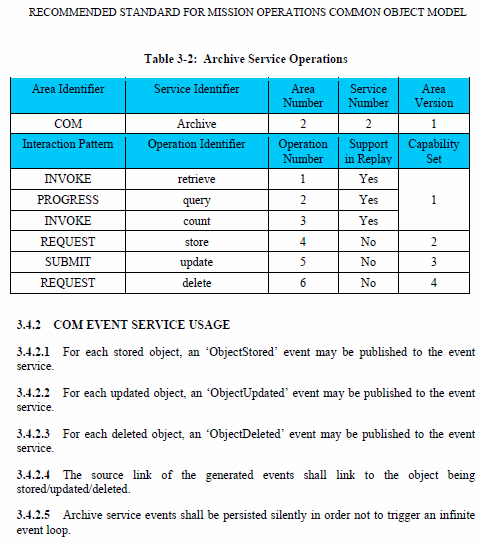
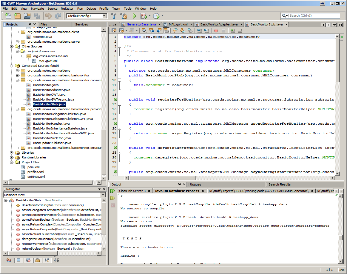
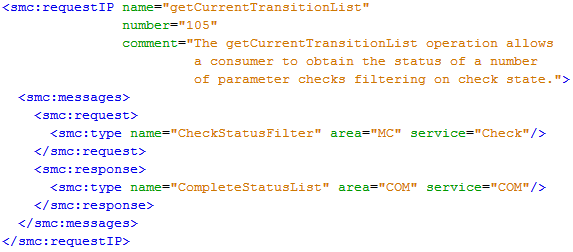


Figure 9 Transformation of MAL into technology-dependent interface specifications

# Analysis

## Areas of Overlap Between the Standards

Components

Encodings

Hardware

Onboard

Mappings

Ground Segment

Interfaces

Types

EDS

MAL

Figure 10 Overlap Between Standards

The two standards have different scopes and purposes, but do have two areas of overlap:

* Interfaces, which describe the set of possible message exchanges between two or more communicating entities.
* Types, which describe and constrain the contents of those messages.

In the case of the hypothetical mission shown in Figure 1, suppose that the hardware device produced by the university has a certain number of configurable settings and modes. The spacecraft OBSW can adjust those settings according to an interface specified in the EDS datasheet for the device.

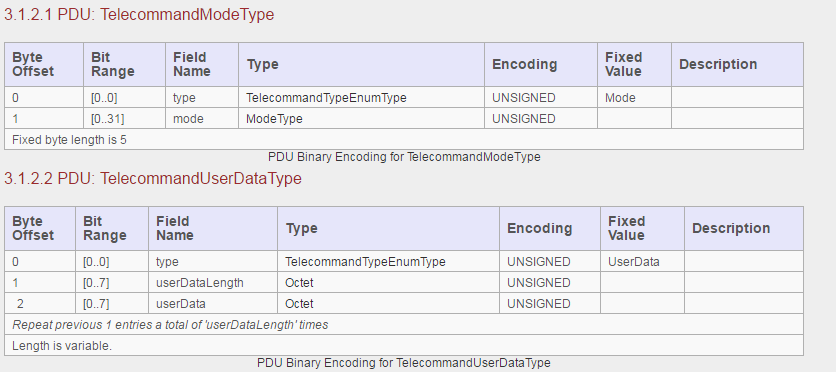


Figure 11 Binary interface to the Device expressed as a CCSDS EDS

The above formatted EDS extract shows how the Protocol Data Units exchanged with the device are split into fields with associated encodings, types and semantics.

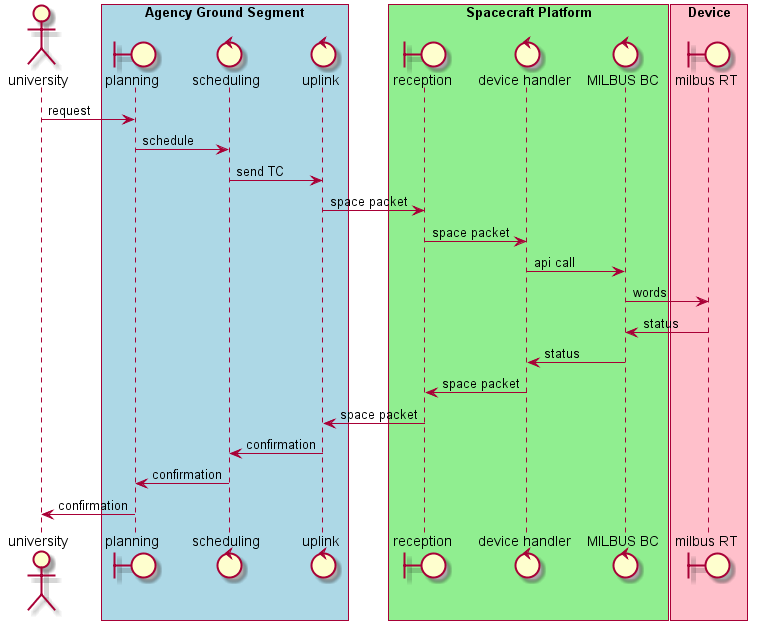


Figure 12 Sequence diagram: adjusting a setting on a device at the request of the end user

The above sequence diagram shows a request being made by an end user, going through a representative range of services on the ground and space, and ultimately resulting in a data exchange of binary words across a MILBUS that conforms to the description specified in a CCSDS EDS.

In effect, a subset of the hardware device interface, as defined in the CCSDS EDS, is made available, via CCSDS MO services, to the end user. Ideally, it would be technically possible to expose every such hardware interface in this way, leaving the decision as to which interfaces *should* be so exposed to be based on the operations concept for the mission.

## Specifying Interfaces

IEEE defines the verb interface as ‘*To connect two or more components for the purpose of passing information from one to the other’.* Thenoun form, ‘an interface’, is a specification of how this is done, exactly what categories of data can be exchanged in what sequences.Between programming languages, standards, middleware tooling, etc. there is a large variety of ways to formally specify an interface. Each such specification makes certain assumptions about what an interface is, in order to describe it.

For our purposes, we can categorise these formalisms according to the following set of properties:

* Message Encoding: how the data in the messages passing across the interface is represented in terms of octets and bits. Can be:
  + Implicit: left to a tool to work out according to a set of defined ‘encoding rules’.
  + Explicit: specified as part of the interface.
  + Optional: a choice of either of the above.
* Cardinality: the number of components connected. Can be 1:1,1:Many or Many:Many
* Directions: From which of the ends of the interface message groups can be initiated. Can be one-way or two-way
* Message Grouping: whether the messages are entirely standalone, or implicitly grouped together by some underlying mechanism. Can be:
  + none: each message is standalone.
  + Paired: each message can have a single reply.
  + Patterned: messages are organised into arbitrarily-large groups according to a set of predefined interaction patterns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Formalism | Terminology | Encoding | Cardinality | Directions | Message Grouping |
| C family[[6]](#footnote-6) | set of functions | implicit[[7]](#footnote-7) | 1:many | one-way | paired[[8]](#footnote-8) |
| PUS[[9]](#footnote-9) | service | explicit | many:many | two-way | none |
| RASDS[[10]](#footnote-10) | port | explicit | 1:1 | two-way | none |
| EDS | interface | optional | 1:1 | two-way | paired |
| MAL | service | implicit | 1:1[[11]](#footnote-11) | one-way | patterned |

Figure 13 Interface Formalism properties

## Detailed Comparative Analysis

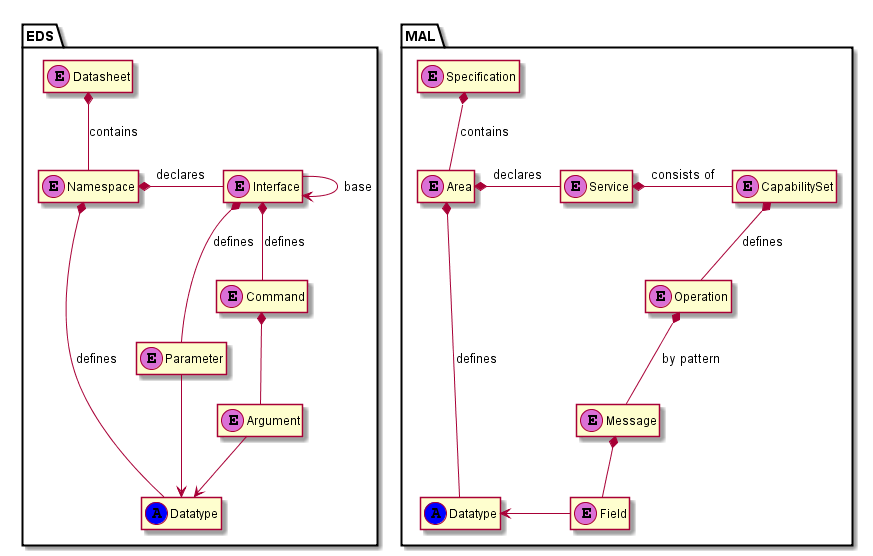


Figure 14 XML structure of EDS and MAL

Areas marked with ‘A’ are abstract, hiding further detail.

When an EDS is used to define an interface:

* A datasheet contains several namespaces.
* Namespaces define data types and interfaces.
* Interfaces use inheritance, and contain parameters and commands.
* Commands have arguments.
* Arguments and parameters have a data type.

When MAL is used to define a service:

* A specification contains several areas.
* Areas define data types and services.
* A service consists of several capability sets.
* Each capability set defines a number of operations
* Each operation has a sequence of message, organised by interaction pattern
* Each message has a number of named fields
* Each field has a data type.

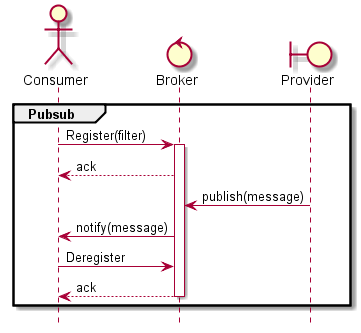
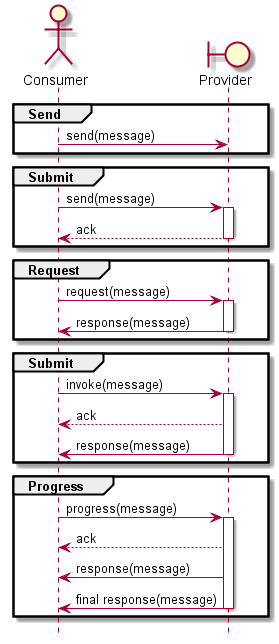


Figure 15 MAL Interaction Patterns

The six supported MAL interaction patterns can be associated with any operation, governing which messages must be specified to define the operation

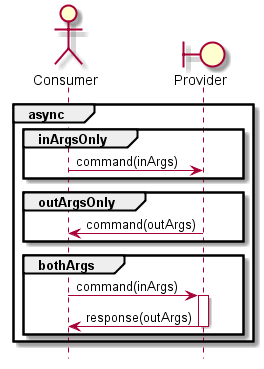
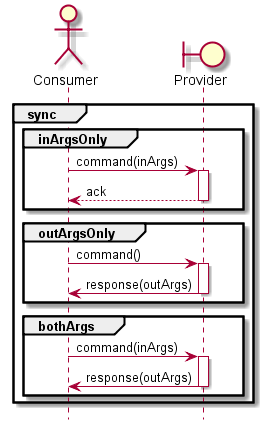


Figure 16 EDS Interaction Patterns

EDS has 4 distinct interaction patterns for commands, based on whether the command mode is async or sync, and whether it has only input arguments, only output arguments, or both.

Three of the EDS interaction patterns map directly to the MAL patterns Send, Submit, and Request. The other, async + outArgsOnly, corresponds to a partial PubSub pattern with no filter.

## Mapping between SOIS EDS and MO MAL

An interface specified in EDS can be mapped to MAL by the following algorithm.

* Within the EDS datasheet:
  + replace each Parameter X with the equivalent list of getX, setX and/or updateX commands, according to the read-only and mode attributes.
  + replace any types defined inline with explicit named type definitions.
* Create a MAL Specification corresponding to the EDS Datasheet.
* For each EDS Namespace involved, create a corresponding MAL Area.
* For each EDS Datatype involved, reference or create a corresponding MAL Datatype.
* Create a MAL Service corresponding to the instantiated Interface specification as used by a particular component.
* Create a MAL Capability Set for each Interface Specification involved in defining that interface.
* Create a MAL Operation for each EDS Command, with interaction pattern set according to:
  + the value of the mode attribute
  + the mode attributes of all arguments to the command.
* Create a MAL Message for each slot in the selected interaction pattern
* Create a MAL Field for each input or output argument of the command, using the matching datatype.

# Prototyping

TODO (if it makes sense and time allows).

# Recommendations and Conclusion

TBW

1. This involvement could potentially take the form of real-time commanding, requests for planning the scheduling of an activity, or be entirely delegated to the agency. In general, the more delegation that occurs, the more information will need to be transferred to the agency in order to support it in making decisions on behalf of the client. [↑](#footnote-ref-1)
2. Supplemented by additional constraints not representable using the XML schema language. [↑](#footnote-ref-2)
3. It is common for there to be no requirement to perform calibration on-board .In such cases the OBSW uses only the access-level interface, while the datasheet still contains calibration data for the sake of ground systems, simulators, etc. [↑](#footnote-ref-3)
4. *Note that SEDS parameters are commonly aggregates of primitive values; as such they arguably more resemble packets than the individual parameters of typical datapool-based software architectures.* [↑](#footnote-ref-4)
5. Commonly this will involves some form of code generation; the principles are the same for manual coding, or even run-time interpretation. [↑](#footnote-ref-5)
6. The programming language C is included because of its historical influence on both other languages like C++ and Java, on middleware targeted at those languages like Corba, RMI and ESA’s SMP2, and also on formalisms designed largely to generate code in such languages, such as UML and SysML Some of those have an explicit ‘interface’ construct corresponding to a set of functions. [↑](#footnote-ref-6)
7. The compiler selects the actual layout of data in memory, according to properties of the target CPU. [↑](#footnote-ref-7)
8. The return value of a function is inherently associated with the corresponding call. [↑](#footnote-ref-8)
9. ESA packet Utilisation Standard, ECSS-E-ST-70-41C. [↑](#footnote-ref-9)
10. Reference Architecture for Space Data Systems (RASDS), CCSDS 311.0-M-1 [↑](#footnote-ref-10)
11. Except a PubSup operation, which has 3 participants. [↑](#footnote-ref-11)