



## 宇宙機情報ベース2定義

(Definition of Spacecraft Information Base 2 (DSIB2))

2023年3月31日 制定

宇宙航空研究開発機構

## 免責条項

ここに含まれる情報は、一般的な情報提供のみを目的としています。JAXA は、かかる情報の正確性、有用性又は適時性を含め、明示又は默示に何ら保証するものではありません。また、JAXA は、かかる情報の利用に関連する損害について、何ら責任を負いません。

## Disclaimer

The information contained herein is for general informational purposes only. JAXA makes no warranty, express or implied, including as to the accuracy, usefulness or timeliness of any information herein. JAXA will not be liable for any losses relating to the use of the information.

## 発行

〒305-8505 茨城県つくば市千現 2-1-1  
宇宙航空研究開発機構 安全・信頼性推進部  
JAXA(Japan Aerospace Exploration Agency)

本書は英語で書かれた草案を日本語に翻訳し、日本の宇宙機関 JAXA により制定された。  
本標準は日本語を正とする。ただし、図表の一部で英語表記しかないものについては、それらが  
正本となる。文章の内容に疑問点がある場合は、日本語及び英語の双方を参照の上、JAXA 安全・  
信頼性推進部まで連絡をすること。

This document was originally drafted in English, then subsequently translated into Japanese and  
authorized by the Japanese space agency, JAXA.

The English translation is for reference purposes only, except for some tables and figures that contain  
English only, in which case they are the original. If there is anything ambiguous about the content of the  
text, please refer to both the Japanese version and the English version and contact JAXA Safety and  
Mission Assurance Department.

# **Definition of Spacecraft Information Base 2 (DSIB2)**

GSTOS 300-1.0

Issue 1.0

31 March 2023

Japan Aerospace Exploration Agency (JAXA)

## 目次

Definition of Spacecraft Information Base 2 (DSIB2).....	i
1. Introduction // はじめに.....	1
1.1. Purpose // 目的 .....	1
1.2. Scope // 範囲 .....	1
1.3. Applicability // 適用先 .....	1
1.4. References // 関連文書 .....	2
1.5. Document Structure // 文書の構造.....	3
1.6. Definitions and Notations // 定義及び表記法.....	4
1.7. Verbal Forms // 表現形式 .....	10
1.8. Conventions // 規則.....	13
2. Overview // 概要 .....	14
2.1. General // 一般 .....	14
2.2. Spacecraft Monitor and Control Protocol (SMCP).....	15
2.3. Generic Spacecraft Test and Operation Software (GSTOS).....	17
2.4. Spacecraft Information Base 2 (SIB2) .....	19
2.5. Telemetry and Telecommand Value Processing // テレメトリ及びテレコマンド 値の処理 .....	23
2.6. Functions of Telecommand Issue SW and Telemetry Monitor SW テ レコマンド発行 SW 及びテレメトリ監視 SW の機能 .....	38
3. Functional/Information Definitions // 機能・情報定義 .....	49
3.1. Overall Structure and Common Definitions // 全体構成及び共通定義 .....	49
3.2. Functional Object Definition // Functional Object 定義 .....	59
3.3. State Machine Definition // State Machine 定義 .....	100
3.4. Attribute Change Rule Definition // Attribute Change Rule 定義 .....	106
3.5. Attribute Limit Definition // Attribute Limit 定義 .....	110
3.6. Condition and Event Definition // Condition and Event 定義 .....	113
3.7. Attribute Sequence Definition // Attribute Sequence 定義.....	116
3.8. Conversion Definition // Conversion 定義.....	127
3.9. Enumeration Definition // Enumeration 定義.....	133
4. Memory Functional Object Definition     Memory Functional Object 定義....	135
4.1. General // 一般 .....	135

4.2. Memory Functional Object .....	136
4.3. Route Definition // 経路定義 .....	137
4.4. Check Sum Definition // Check Sum 定義 .....	145
5. Naming/ID assignment rules // 命名・ID 付与規則 .....	146
5.1. General // 一般 .....	146
5.2. Rules for Defining Names // 名前を定義する規則 .....	146
5.3. Rules for assigning IDs // IDs 割り当て規則 .....	149
5.4. Telemetry Item Name, Telecommand Name, and Command Name Telemetry 項目名、Telecommand 名、及び Command 名 .....	150
6. Diagnostic Rule Definition // Diagnostic Rule 定義 .....	153
6.1. General // 一般 .....	153
6.2. Diagnostic Rule .....	153
7. Other Subsidiary Definitions // その他の副次的な定義 .....	154
7.1. Spacecraft Information Definition // 宇宙機情報定義 .....	154
7.2. MAP ID Definition // MAP ID 定義 .....	155
7.3. Telemetry Packet Length Information Definition // テレメトリパケット長情報 定義 .....	156
8. APPENDIX A. ACRONYMS // 略語 .....	157

## 1. Introduction // はじめに

### 1.1. Purpose // 目的

[Generic Spacecraft Test and Operation Software \(GSTOS\)](#) [A8] is software used for generic purposes in tests and operations of a spacecraft and its onboard instruments. The [Spacecraft Information Base 2 \(SIB2\)](#) [A8] is a database for storing information on spacecraft (*i.e.*, artificial satellites and space probes) and their onboard instruments. This document specifies a set of information stored in the [SIB2](#) for use in the [GSTOS](#).

With respect to the information items defined in this document, the data stored in [SIB2](#) should be used as the sole source data.

### 1.2. Scope // 範囲

This document is defined in a technology-independent format that handles the structure (data model) of the information handled with [SIB2](#). It does not specify the specific format to be saved in files, the specification of the application that handles [SIB2](#), or how to implement such an application.

### 1.3. Applicability // 適用先

This document applies to the specification of subsystems and instruments onboard spacecraft and their ground support equipment all of which conform to [SIB2](#).

[Generic Spacecraft Test and Operation Software \(GSTOS\)](#) [A8] は、宇宙機やその搭載機器の試験・運用で汎用に使用するソフトウェアである。[Spacecraft Information Base 2 \(SIB2; 宇宙機情報ベース2\)](#) [A8] は、宇宙機（つまり、人工衛星や宇宙探査機）とその搭載機器に関する情報を格納するためのデータベースである。本書は、[GSTOS](#)において用いる [SIB2](#) に格納される一連の情報を規定する。

本書で規定する情報項目に関しては、[SIB2](#) に格納されたデータを唯一の源泉データとして用いるべきである。

この文書は、[SIB2](#) で扱う情報の構造（データモデル）を扱う技術に依存しない形式で定める。ファイルに保存する具体的な形式や、[SIB2](#) を扱うアプリケーションの仕様やそういったアプリケーションを如何に実装するかは定めない。

この文書は、[SIB2](#) に準拠する、宇宙機搭載のサブシステムと機器、並びに、それらの地上支援装置の仕様に適用する。

## 1.4. References // 関連文書

### 1.4.1. Normative References // 引用文書

- [A1] JAXA, “Standard of Communications and Data-Handling Architecture, Part 1: General, Part 5: Onboard Subnetwork Protocol Architecture, Part 6: Ground Subnetwork Protocol Architecture (SCDHA156),” SCDHA 110-1.1, JERG-2-400-TP101, December 2019.
- [A2] JAXA, “Functional Model of Spacecrafts (FMS)”, GSTOS 201-1.1, JERG-2-700-TP001 (NOTICE-1), March 2020 (March 2023).
- [A3] JAXA, “Spacecraft Monitor and Control Protocol (SMCP)”, GSTOS 200-1.1, JERG-2-700-TP002 (NOTICE-1), December 2019 (March 2023).
- [A4] JAXA, “Standard of Communications and Data-Handling Architecture, Part 2: End-to-end Protocol Architecture (SCDHA2)”, SCDHA 120-1.0, JERG-2-400-TP102, December 2019.
- [A5] CCSDS, “Space Packet Protocol”, CCSDS 133.0, latest issue.
- [A6] CCSDS, “TC Space Data Link Protocol”, CCSDS 232.0, latest issue.
- [A7] CCSDS, “Communications Operation Procedure-1”, CCSDS 232.1-B-2, September 2010.
- [A8] JAXA, “Generic Spacecraft Test and Operation Software (GSTOS) Requirement Specification (GRS)”, GSTOS 400, JERG-2-700-TP003 (TBD), latest issue.

### 1.4.2. Informative References // 参考文書

- [R1] W3C, “Extensible Markup Language (XML) 1.0”, Second Edition, 6 October 2000.
- [R2] JAXA, 「SIB2 XML Schema 定義書」、GSTOS 311, RDA-2018002, 最新版
- [R3] JAXA, 「SIB2 メモリ管理ファイル仕様書」、RDA-14021, 最新版
- [R4] 「衛星自動監視ソフトウェア取扱説明書」、最新版
- [R5] JAXA, 「SIB2 システム仕様書」、RDA-2018021, 最新版
- [R6] 「SIB2 作成ツール (SIB2UI) 入力リファレンス」、最新版
- [R7] 「SIB2 メモリ管理表」、最新版

## 1.5. Document Structure // 文書の構造

This document is organized as follows.

Chapter 1 (this chapter) states the purpose, scope, and applicability of the document, and lists the references, definitions, and notations used throughout the document.

Chapter 2 presents an overview of the [GSTOS](#) and [SIB2](#).

Chapter 3 specifies the [Function/Information Definitions](#), which are used to specify the design of [Functional Objects](#) and [SMCP Telecommands](#) and [SMCP Telemetries](#) according to the [Functional Model of Spacecrafts \(FMS\)](#) [A2] and the [Spacecraft Monitor and Control Protocol \(SMCP\)](#) [A3].

Chapter 4 specifies the [Memory Functional Object Definitions](#), which define the design parameters of onboard memories.

Chapter 5 specifies the rules for names and identifiers in the [Function/Information Definitions](#) and [Memory Functional Object Definitions](#).

Chapter 6 specifies the [Diagnostic Rule Definitions](#), which are used to diagnose whether [Functional Objects](#) are functioning correctly or not.

Chapter 7 specifies the other subsidiary definitions.

Appendix A lists the acronyms used in this document.

本書は次の通り構成する。

1章（本章）は、本書の目的、範囲及び適用先を述べると共に、本書で用いる関連文書、定義、及び表記法を示す。

2章は、[GSTOS](#)と[SIB2](#)を概説する。

3章は、[Functional Model of Spacecrafts \(FMS\)](#) [A2] 及び [Spacecraft Monitor and Control Protocol \(SMCP\)](#) [A3] に従って、[Functional Objects](#)や[SMCP Telecommands](#)及び[SMCP Telemetries](#)の設計を規定するのに用いるものである [機能・情報定義](#)を規定する。

4章は、搭載メモリの設計パラメータを定義するものである [Memory Functional Object 定義](#)を規定する。

5章は、[機能・情報定義](#)及び[Memory Functional Object 定義](#)における命名・ID付与規則を規定する。

6章は、[Functional Objects](#)が正常に動作しているかどうかを診断するのに用いるものである [Diagnostic Rule 定義](#)を規定する。

7章は、その他の副次的な定義を規定する。

Appendix Aは、本書で用いる略語を示す。

## 1.6. Definitions and Notations // 定義及び表記法

### 1.6.1. Terms defined in the SCDHA Part 1 // SCDHA Part 1 で定義される用語

This document adopts the following terms defined in the “Standard of Communications and Data-Handling Architecture, Part 1: General (SCDHA1)” [A1]:

Space science project (or simply project).

本書では、“Standard of Communications and Data-Handling Architecture, Part 1: General (SCDHA1)” [A1]で定義される次の用語を採用する。

Space science project (または単に、project)

### 1.6.2. Terms defined in the Functional Model of Spacecrafts

#### Functional Model of Spacecrafts で定義される用語

This document adopts the following terms defined in the “Functional Model of Spacecrafts (FMS)” [A2]:

Functional Object, (Parent Functional Object // 親 Functional Object), (Child Functional Object // 子 Functional Object), Attribute, Initial Value, Binary Sequence Attribute, Enumerative Attribute, Valid Value Set, State Attribute, Numerical Value Attribute, Action Limit, Caution Limit, Attribute Change Rule, Operation, Criticality Level, Parameter, Valid Range, Alert, Alert class, Event, Event class, State Machine, State, Initial State, Begin State, End State, State Transition class, Maximum Allowable Transition Time, Minimum Allowable Transition Time, Trigger class, Trigger Condition, (Effective-Condition // 有効条件), (Condition Expression // 条件式), Comparison Term, Diagnostic Rule, Memory Functional Object, FirstAddress, LastAddress, MaximumUploadLength, and AlignmentLength.

### 1.6.3. Terms defined in the Spacecraft Monitor and Control Protocol Spacecraft Monitor and Control Protocol で定義される用語

This document adopts the following terms defined in the “Spacecraft Monitor and Control Protocol (SMCP)” [A3]:

Controller, Monitor, Attribute Sequence, SMCP Message, SMCP Telecommand, Telecommand Message, ACTION Telecommand, GET Telecommand, SET Telecommand, MEMORY LOAD Telecommand, MEMORY DUMP Telecommand, SMCP Telemetry, Telemetry Message, VALUE Telemetry, Response VALUE Telemetry, Acknowledgement Telemetry (ACK Telemetry), NOTIFICATION Telemetry, MEMORY DUMP Telemetry, Attribute Identifier (Attribute ID), Alert Identifier (Alert ID), Operation Identifier (Operation ID), Upper FOID, Lower FOID, Route Identifier (Route ID), Upper Functional Object and Route Identifier (UFORID), Attribute Values field, and Parameters and Attribute Values field.

### 1.6.4. Terms defined in the SCDHA Part 2 // SCDHA Part 2 で定義される用語

This document adopts the following terms defined in the “Standard of Communications and Data-Handling Architecture, Part 2: End-to-End Protocol (SCDHA2)” [A4]:

Lower APID and

Upper APID.

本書では、“Spacecraft Monitor and Control Protocol (SMCP)” [A3] で定義される次の用語を採用する。

本書では、“Standard of Communications and Data-Handling Architecture, Part 2: End-to-End Protocol (SCDHA2)” [A4] で定義される次の用語を採用する。

Lower APID

Upper APID

### 1.6.5. Terms defined in the Space Packet Protocol // Space Packet Protocol で定義される用語

This document adopts the following terms defined in the “Space Packet Protocol” [A5]:

Application Process Identifier (APID) and  
Space Packet.

本書では、“Space Packet Protocol” [A5] で定義される次の用語を採用する。

Application Process Identifier (APID)  
Space Packet

### 1.6.6. Terms defined in the TC Space Data Link Protocol // TC Space Data Link Protocol で定義される用語

This document adopts the following terms defined in the TC Space Data Link Protocol (TC SDLP) [A6]:

MAP Channel,

Multiplexer Access Point Identifier (MAP ID), and

TC Transfer Frame.

本書では、TC Space Data Link Protocol (TC SDLP) [A6] で定義される次の用語を採用する。

MAP Channel

Multiplexer Access Point Identifier (MAP ID)

TC Transfer Frame

### 1.6.7. Terms defined in the GSTOS Requirements Specification

#### GSTOS Requirements Specification で定義される用語

This document adopts the following terms defined in the GSTOS Requirements Specification (GRS) [A8]:

Telecommand Issue Software (Telecommand Issue SW),

Generic Spacecraft Test and Operations Software (GSTOS),

Spacecraft Information Base 2 (SIB2),

and

Telemetry Monitor Software (Telemetry Monitor SW).

Telecommands are transferred from Telecommand Issue SW to a spacecraft and then to the onboard instruments that execute the telecommands. In a telecommand transfer, either Telecommand Issue SW or one of the onboard instruments is responsible to control the execution timing of the telecommand and might hold the telecommand for a length of time if required. The transfer processes in any other sections are performed without delay. In this document, "to control the timing at which the instrument executes a telecommand by sending the telecommand" is expressed as "to issue a telecommand" according to the GRS.

本書では、GSTOS Requirements Specification (GRS) [A8] で定義される次の用語を採用する。

テレコマンド発行ソフトウェア（テレコマンド発行 SW）

Generic Spacecraft Test and Operations Software (GSTOS)

Spacecraft Information Base 2 (SIB2; 宇宙機情報ベース 2)

テレメトリ監視ソフトウェア（テレメトリ監視 SW）

テレコマンドはテレコマンド発行 SW から宇宙機に伝送され、さらに、テレコマンドを実行する搭載機器まで送り届けられる。テレコマンドの伝送において、テレコマンド発行 SW か宇宙機上の何れかの装置がテレコマンドを必要に応じある時間留め置く事があり、その実行タイミングの決定に責任を持つ。この装置以外の伝送途中有る装置はテレコマンドを遅滞なく伝送する。本書では GRS に従って「あるテレコマンドを送信する事により、機器がそのテレコマンドを実行するタイミングを司る」事を「テレコマンドを発行する」と表現する。

### 1.6.8. Terms defined in this document // 本書で定義される用語

The following definitions are used throughout this document. 本書は次の定義を用いる。

#### Absolute Name:

the name that uniquely identifies something in an entire spacecraft.

#### 絶対名:

あるものを宇宙機全体で一意に識別する名前。

#### Local Name:

the name that uniquely identifies something that belongs directly to another.

#### 局所名:

あるものに直接属するものを一意に識別する名前。

#### Reference Name:

a name used for referencing (either an **Absolute Name** or **Relative Name**)

#### 参照名:

参照に用いる名前（**絶対名**か**相対名**の何れか）

#### Relative Name:

the name that uniquely identifies something referencing from a **Functional Object**.

#### 相対名:

ある **Functional Object** を起点とし、他のものを一意に識別する名前。

### 1.6.9. Notations // 表記

The following notations are used throughout this document.

A paragraph that begins with “[Example]” (or “[Example *n* ]”, where *n* is a positive integer) presents an example that is aimed to help readers to understand the specification, and is not a part of the specification.

A paragraph that begins with “[Note]” (or “[Note *n* ]”, where *n* is a positive integer) contains an informative note that is aimed to help readers to understand the specification, and is not a part of the specification.

A paragraph that begins with “[Rationale]” (or “[Rationale *n* ]”, where *n* is a positive integer) contains a rationale for the specification, but is not a part of the specification.

A rationale may be simply a phrase or a (series of) complete sentence(s). In the former case, the phrase usually starts with a lower-case letter and it should be interpreted as if it was preceded with an implicit partial sentence of “The author of this document sets the specification as such”. For example, “[Rationale] in order to distinguish clearly X and Y.” means that the author of this document sets the specification as such in order to distinguish clearly X and Y.”

次の表記を本書において適用する。

“[例]”（または“[例 *n* ]”、*n* は正の整数）で始まる段落は、読者の仕様の理解を助けるための例であり、仕様の一部ではない。

“[注]”（または“[注 *n* ]”、*n* は正の整数）で始まる段落は、読者の仕様の理解を助けるための付加情報を記したものであり、仕様の一部ではない。

“[根拠]”（または“[根拠 *n* ]”、*n* は正の整数）で始まる段落は、仕様の根拠を記したものであり、仕様の一部ではない。

## 1.7. Verbal Forms // 表現形式

The following conventions apply throughout this document.

- a) the auxiliary verb ‘**shall**’ implies mandatory conditions.
- b) the auxiliary verb ‘**should**’ implies optional but desirable conditions.
- c) the auxiliary verbs ‘**may**’ implies optional conditions.
- d) the auxiliary verb ‘can’ implies capability or ability to do something.
- e) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

The words ‘**shall**’, ‘**should**’, ‘**may**’ are highlighted in **red** and **bold** font.

本書では以下の決まりに従い記述する。

- a) 「...**こと**」「...**なければならない**」は、必須な仕様を示す。
- b) 「...**べき**...」は、任意であるが推奨される仕様を示す。
- c) 「...**良い**...」は、許容される仕様を示す。
- d) 「...できる...」は、何かをする事が可能な事を示す。
- e) 他のパターンの記述は、事実を示す文である。

「...**こと**」「...**なければならない**」「...**べき**...」「...**良い**...」は読者の仕様の理解の助けのため、**赤字・太字**で示す。

[注] 本書では、要求事項を電子的に検索しやすいように、英文の‘**shall**’の訳語として、「**こと**」を使用している。逆に、‘**shall**’の訳語以外では「こと」は使用せず、「事」を用いている。また、英文の‘**may**’に対応する訳語として、「**良い**」という当て字を使用している。逆に、‘**may**’の訳語以外で「良い」は使用していない。

「A, B, 及び C」という表記は、英文の ‘A, B, and C’ に対応し、「A 及び B 及び C」である事を意味する。

「A, B, または C」という表記は、英文の ‘A, B, or C’ に対応し、「A または B または C」である事を意味する。

When a translation into Japanese is provided, the original English version and its Japanese translation are given in the left and right sides, respectively, in principle, as in this paragraph. In some cases, *e.g.*, titles of sections and captions of figures/tables, the English and Japanese versions are put in a single line separated by “//” in this order (“English // Japanese”) or in separate lines with no delimiter in between (“English [Line-Break] Japanese”).

In most cases, the technical terms are not translated into Japanese. The English words in alphabet remain as they are in their Japanese translations. The forms in alphabet in English, which distinguish the singular and plural forms, remain as they are in the Japanese translations to preserve the information of the quantity, although the Japanese language does not inherently distinguish the singular and plural forms.

日本語への翻訳が存在する場合、原則として、この段落のように、英語を左側に示し、日本語を右側に示す。また、章や図表のタイトル等は、英語、日本語の順に一行中に // で区切る（「英語 // 日本語」）か、二行に分けて区切り文字なし（「英語 [改行] 日本語」）で、記述する場合もある。

多くの場合、技術用語の翻訳は行わず、英単語を維持する。そこで、日本語にもアルファベットが登場する。それらは正本である日本語文中においてもアルファベット表記される。日本語の名詞に単数形、複数形の区別はないが、單複の情報を保つため、日本語文中においても、英語の単数形、複数形の違いはアルファベットでそのまま表記する。

Technical terms are highlighted in **green** and in some cases (names of states and those defined in the other documents) in **blue**. The latter consists of names of documents, protocols, widely used technical terms, and those locally used in some sections (*e.g.*, field names). Note that the first letter of an English word in a technical term is written in the capital letter excluding that in a widely used technical term.

A technical term is underlined as “Term” in each of their first occurrences in chapters 1, 2, and 3 onwards.

References are written in one of the following formats in this document; *n.b.*, “a word” here includes a compound word, for example, “MEMORY LOAD Telecommand”.

- 1) a word (Section X.Y.Z)
- 2) a phrase or sentence (see Section X.Y.Z)
- 3) a reference to a definition item [Section X.Y.Z WWW Definition n)]

技術用語は読者の便のため**緑字**、場合（状態の名前と他の文書で定義された用語）により**青字**で示す。後者は、文書名、プロトコル名、広く用いられている技術用語、及び、局所的にしか登場しないもの（フィールド名等）からなる。ここで、技術用語は、広く用いられているものを除き、基本的に大文字始まりの英単語で表記する。

技術用語は、1章、2章、3章以降の各々の初出箇所を「用語」のように下線で明示する。

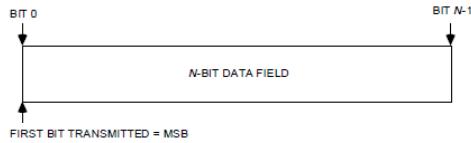
本書では、参照を、以下の形式の何れかで記述する（注：ここで「**単語**」には、“**MEMORY LOAD Telecommand**”などの複合語が含まれる）。

- 1) 単語 (X.Y.Z 項)
- 2) 句または文 (X.Y.Z 項参照)
- 3) 定義項目への参照 [X.Y.Z 項 WWW Definition n)]

## 1.8. Conventions // 規則

In this document, the following conventions are used to identify each bit in an **N-bit** field. The first bit in the field to be transmitted (*i.e.*, the leftmost part in associated diagrams, if given) is defined as '**Bit 0**', the next bit is defined as '**Bit 1**', and so on up to '**Bit N-1**'.

本書では、**N-bit** のフィールドの各ビットを識別するため、次の規則を用いる。フィールドの中で伝送する最初のビット（つまり、図示する場合、図中の最も左側）を'**Bit 0**'と定義する。以下、次のビットを'**Bit 1**'というように'**Bit N-1**'まで定義する。



**Figure 1-1: Bit Numbering Convention  
Bit 番号付け規則**

When a field is used to express a binary value (such as a counter), the Most Significant Bit (**MSB**) **shall** be the first transmitted bit of the field, *i.e.*, '**Bit 0**' (see Figure 1-1).

In accordance with the standard data-communications practice, data fields are often grouped into a series of eight-bit 'words'. Throughout this document, this unit of an eight-bit word is referred to as an 'octet'.

フィールドがバイナリ値（カウンタ等）を表わす場合、最上位ビット (**MSB**) はフィールドの最初に伝送するビット、つまり、Figure 1-1 に示す '**Bit 0**' であること。

標準的なデータ通信の慣例に則り、データフィールドは、しばしば、8 ビットワードの連なりにまとめる。本書では、この 8 ビットワードの単位を 'octet' と称する。

## 2. Overview // 概要

### 2.1. General // 一般

The [Standard of Communications and Data-Handling Architecture \(SCDHA\)](#), Normative Reference [A1]) specifies the standard framework for the onboard and ground systems for communications/data-handling that are used in spacecrafts for science missions developed by [space science projects](#). This model sets a set of standardized methods to specify functions of any spacecrafts and to manage electronically information of the functions. This standardized model would make systematic development of spacecraft functions easier and make reusing the existing onboard instruments or parts of them practical. Then, the ultimate purpose is to reduce the cost of development of new spacecrafts and to enhance their reliability

[Standard of Communications and Data-Handling Architecture \(SCDHA](#), 引用文書 [A1] 参照) は、[space science projects](#) が開発する科学ミッション等のための、衛星搭載及び地上の 通信・データハンドリングシステムの開発に用いられる標準的な枠組みを定める。このモデルは衛星の機能を定め、その機能の情報を電子的に管理する標準化された一群の手法を与える。この標準化されたモデルは、衛星の機能を系統的に開発する事を容易にすると共に、既存の衛星搭載機器やその一部の再利用を現実的なものとする。これらの究極的な目的は、新たな衛星の開発コストを削減し、信頼性向上する事にある。

The [SCDHA](#) assumes that the functions of the relevant components in a spacecraft are specified according to the [Functional Model of Spacecrafts \(FMS\)](#) [A2]. We note that the [SCDHA](#) assumes that the [Spacecraft Monitor and Control Protocol \(SMCP\)](#) [A3] is adopted for monitoring and controlling a spacecraft and its onboard instruments by the spacecraft operation systems located on the ground and onboard the spacecraft. The individual specifications according to the model/protocol are supposed to be stored and managed by the [Spacecraft Information Base 2 \(SIB2\)](#).

[SCDHA](#) は、宇宙機の関連機器の機能が、[Functional Model of Spacecrafts \(FMS\)](#) [A2] に従って定められている事を前提としている。また、[SCDHA](#) は、宇宙機やその搭載機器を、地上及びその宇宙機搭載の宇宙機運用システムが監視制御するために、[Spacecraft Monitor and Control Protocol \(SMCP\)](#) [A3] を採用する事を前提としている。また、本モデル・プロトコルに従う個々の仕様は、[Spacecraft Information Base 2 \(SIB2\)](#) によって蓄積、管理する事が想定されている。

## 2.2. [Spacecraft Monitor and Control Protocol \(SMCP\)](#)

In [SMCP](#), telecommands are categorized into the following five types.

- a) [ACTION Telecommand](#) (Section 3.2.5.2)
- b) [GET Telecommand](#) (Sections 3.2.3.2 and 3.7.2)
- c) [SET Telecommand](#) (Sections 3.2.3.2 and 3.7.2)
- d) [MEMORY LOAD Telecommand](#) (Section 4.3.2.2)
- e) [MEMORY DUMP Telecommand](#) (Section 4.3.2.3)

In [SIB2](#), each of these is defined in the manner noted in the references above.

[SMCP](#) では、テレコマンドは、以下の五つのタイプに分類される。

- a) [ACTION Telecommand](#) (3.2.5.2 項)
- b) [GET Telecommand](#) (3.2.3.2 項及び 3.7.2 項)
- c) [SET Telecommand](#) (3.2.3.2 項及び 3.7.2 項)
- d) [MEMORY LOAD Telecommand](#) (4.3.2.2 項)
- e) [MEMORY DUMP Telecommand](#) (4.3.2.3 項)

[SIB2](#) では、これらをそれぞれ上記の参照に示す方式で定義する。

In **SMCP**, telemetries are categorized into the following four types.

- a) VALUE Telemetry (Sections 3.2.3.2 and 3.7.2)
- b) NOTIFICATION Telemetry (Section 3.2.6.2)
- c) Acknowledgement Telemetry (ACK Telemetry) (Section 3.2.2.1)
- d) MEMORY DUMP Telemetry (Section 4.3.3)

In **SIB2**, each of these is defined in the manner noted in the references above.

A **VALUE Telemetry** or **SET Telecommand** that contains only one sub-field for one **Attribute** in the **Attribute Values** field is defined in the manner noted in Section 3.2.3.2 (or Section 3.7.2).

A **VALUE Telemetry** or **SET Telecommand** that contains one or more sub-fields for **Attributes** in the **Attribute Values** field is defined in the manner noted in Section 3.7.2.

**SMCP** では、テレメトリーは、以下の四つのタイプに分類される。

- a) VALUE Telemetry (3.2.3.2 項及び 3.7.2 項)
- b) NOTIFICATION Telemetry (3.2.6.2 項)
- c) Acknowledgement Telemetry (ACK Telemetry) (3.2.2.1 項)
- d) MEMORY DUMP Telemetry (4.3.3 項)

**SIB2** では、これらをそれぞれ上記の参照に示す方式で定義する。

**VALUE Telemetry** または **SET Telecommand** で **Attribute Values** フィールドに、一つの **Attribute** に対するサブフィールドを一つのみ含むものは 3.2.3.2 項 (もしくは 3.7.2 項) に示す方式で定義する。

**VALUE Telemetry** または **SET Telecommand** で **Attribute Values** フィールドに、**Attributes** に対するサブフィールドを一つ以上含むものは 3.7.2 項に示す方式で定義する。

### 2.3. Generic Spacecraft Test and Operation Software (GSTOS)

Generic Spacecraft Test and Operation Software (GSTOS) is software used for generic purposes in tests and operations of a spacecraft and its onboard instruments. GSTOS includes Telecommand Issue Software (hereafter, Telecommand Issue SW) and Telemetry Monitor Software (hereafter, Telemetry Monitor SW). Telecommand Issue SW is used to issue telecommands transferred from the ground to a spacecraft and to manage the onboard memories. Telemetry Monitor SW is used to monitor telemetries from the spacecraft to the ground.

GSTOS assumes that a spacecraft and its onboard instruments are designed according to the SCDHA.

The SCDHA assumes that the functions of the relevant components in a spacecraft are specified according to the FMS. The SCDHA also assumes that the Spacecraft Monitor and Control Protocol (SMCP) [A3] is adopted for monitoring and controlling a spacecraft and its onboard instruments by the spacecraft operation systems located on the ground and onboard the spacecraft.

Telecommand Issue SW issues telecommands to a spacecraft.

Generic Spacecraft Test and Operation Software (GSTOS) は、宇宙機やその搭載機器の試験・運用で汎用に使用するソフトウェアである。GSTOS は、テレコマンド発行ソフトウェア（以降、テレコマンド発行 SW）とテlemetry監視ソフトウェア（以降、テlemetry監視 SW）を含む。テレコマンド発行 SW は、地上から宇宙機へ伝送されるテレコマンドを発行すると共に、搭載メモリの管理に用いる。テlemetry監視 SW は、宇宙機から地上へのテlemetryの監視に用いる。

GSTOS は、宇宙機やその搭載機器が SCDHA に従って設計される事を想定している。

SCDHA は、宇宙機の関連機器の機能が、FMS に従って定められている事を前提としている。また、SCDHA は、宇宙機やその搭載機器を、地上及びその宇宙機搭載の宇宙機運用システムが監視制御するために、Spacecraft Monitor and Control Protocol (SMCP) [A3] を採用する事を前提としている。

テレコマンド発行 SW は、宇宙機へのテレコマンドを発行する。

Commands to [Telecommand Issue SW](#) are categorized into two types: ones to issue a telecommand and ones to control the operation of [Telecommand Issue SW](#), referred to as a [Telecommand Issue SW Control Command \(CISW Control Command\)](#). [Telecommand Issue SW](#) issues telecommands to a spacecraft on an execution of a command to [Telecommand Issue SW](#).

[Telecommand Issue SW](#) and [Telemetry Monitor SW](#) have the common specifications that are allocated to the module referred to as a [Data Conversion Module](#). The common specifications include [Binary Sequence Extraction/Embedding](#), [Numerical Decoding/Encoding](#), [Engineering Unit Conversions](#), and [String Conversion](#) (Sections 2.5.3.4, 2.5.4.3, 2.5.5.2, and 2.5.6.2, respectively).

テレコマンド発行 SW へのコマンドは、二つの種類: テレコマンドを発行するためのもの、[テレコマンド発行 SW 制御コマンド \(CISW 制御コマンド\)](#) と称するテレコマンド発行 SW の動作を制御するためのものに分類される。テレコマンド発行 SW は、テレコマンド発行 SW へのコマンドを実行により宇宙機へのテレコマンドを発行する。

テレコマンド発行 SW とテレメトリ監視 SW は、[データ変換モジュール](#)と称するモジュールに割り当てられる共通の仕様を持つ。共通の仕様として [バイナリ列抽出・埋込](#)、[数値デコード・エンコード](#)、[工学値変換](#)、及び[文字列変換](#)（それぞれ、2.5.3.4 項、2.5.4.3,項、2.5.5.2 項、及び 2.5.6.2 項）などがある。

## 2.4. Spacecraft Information Base 2 (SIB2)

The telemetry and telecommand database called the [Spacecraft Information Base 2 \(SIB2\)](#), which this document specifies, stores and manage the functional and operational design of a spacecraft and its onboard instruments (both specified according to the [FMS](#)) and the design of the telecommands and telemetries (both specified according to the [SMCP](#)).

[GSTOS](#) is configured to handle a spacecraft or an onboard instrument according to the information of the [SIB2](#).

[SIB2](#) consists of the [Function/Information Definition](#), [Memory Functional Object Definition](#), [Diagnostic Rule Definition](#), and the other subsidiary definitions.

The [Function/Information Definition](#) defines the parameters with respect to [Functional Objects](#), *i.e.*, the design parameters of the [Functional Objects](#) and the configuration parameters for the functions of [Telecommand Issue SW](#) and [Telemetry Monitor SW](#) (see Chapter 3).

The [Function/Information Definition](#) consists of the [Function Definition](#) and the [Information Definition](#).

本書が規定する [Spacecraft Information Base 2 \(SIB2; 宇宙機情報ベース 2\)](#) と呼ばれるテレメトリ及びテレコマンドのデータベース（テレコマデータベース）は、([FMS](#) に従い定められた) 宇宙機やその搭載機器の機能・運用設計、並びに、([SMCP](#) に従い定められた) テレコマンドとテレメトリの設計を蓄積、管理する。

[GSTOS](#) は、[SIB2](#) の情報に従って、特定の宇宙機または搭載機器を扱うように設定される。

[SIB2](#) は、[機能・情報定義](#)、[Memory Functional Object 定義](#)、[Diagnostic Rule 定義](#)、及びその他の副次的な定義からなる。

[機能・情報定義](#)は、[Functional Objects](#) に関する、それらの設計パラメータ、並びに、[テレコマンド発行 SW](#) 及び[テレメトリ監視 SW](#)の機能に関する設定パラメータを定義する（3章参照）。[機能・情報定義](#)は、[機能定義](#)と[情報定義](#)からなる。

The **Function Definition** defines the functions of a spacecraft or onboard instruments using the concept of the **Functional Object** as defined in the **FMS**. For each **Functional Object**, the **Function Definition** defines the **Operations** executed by the **Functional Object**, the **Attributes** representing the status of the **Functional Object**, the **State Machines** of the **Functional Object**, etc.

The **Information Definition** defines the data format of **SMCP Messages** (*i.e.*, **SMCP Telecommands** to control **Functional Objects** and **SMCP Telemetries** to monitor **Functional Objects**) specified in the **Spacecraft Monitor and Control Protocol (SMCP)** [A3] for monitoring and controlling **Functional Objects**.

[Note] Since the current version of **SMCP** does not support the use of shared memory in the transmission of **SMCP Messages**, the definition of the correspondence between an **SMCP Message** and a memory is not included in this version of **SIB2**.

The **Function/Information Definition** is a collection of the following definitions.

- 1) **Functional Object Definition**
- 2) **State Machine Definition**
- 3) **Attribute Change Rule Definition**
- 4) **Attribute Limit Definition**
- 5) **Condition and Event Definition**
- 6) **Attribute Sequence Definition**
- 7) **Conversion Definition**
- 8) **Enumeration Definition**

**機能定義**は、宇宙機または搭載機器の機能を **FMS** で規定されている **Functional Object** の概念を用いて定義する。**Functional Object** 每に、その **Functional Object** が実行する **Operations**、その **Functional Object** の状態を表す **Attributes**、**Functional Object** の **State Machines** 等を**機能定義**にて定義する。

**情報定義**は、**Functional Objects** を監視制御するために **Spacecraft Monitor and Control Protocol (SMCP)** [A3]で規定されている **SMCP Messages**(すなわち、**Functional Objects** を制御するための **SMCP Telecommands** と **Functional Objects** を監視するための **SMCP Telemetries**) のデータフォーマットを定義する。

[注] 現バージョンの **SMCP** は、**SMCP Messages** の伝送において共有メモリの使用はサポートしていないため、**SMCP Message** とメモリとの対応関係の定義は、**SIB2** の本版には含まれていない。

**機能・情報定義**は、以下の定義の集合である。

- 1) **Functional Object 定義**
- 2) **State Machine 定義**
- 3) **Attribute Change Rule 定義**
- 4) **Attribute Limit 定義**
- 5) **Condition and Event 定義**
- 6) **Attribute Sequence 定義**
- 7) **Conversion 定義**
- 8) **Enumeration 定義**

The method of expressing the **Function/Information Definition** by means of a set of files in the **XML** format [R1] called **SIB2XML** is specified in Informative Reference [R2].

In **SIB2XML**, each of the above-listed definitions is represented in a single file. Both applications that create the **Function/Information Definition** and applications that use the **Function/Information Definition** access **SIB2XML**.

The **Memory Functional Object Definition** defines the parameters with respect to the **Memory Functional Objects**, *i.e.*, the design parameters of the **Memory Functional Objects** and the configuration parameters for the functions of **Telecommand Issue SW** (see Chapter 4).

The method of expressing the **Memory Functional Object Definition** by means of an **XML**-format file called a **Memory Management File** is specified in Informative Reference [R3].

The **Diagnostic Rule Definition** defines **Diagnostic Rules**, which determine whether a **Functional Object** is operating properly (see Chapter 6). Note that **Automated Monitoring Software (ATMOS)** can be used instead to define diagnostic rules that are much more complex than the **Diagnostic Rule Definition** (see Informative Reference [R4]).

**機能・情報定義を SIB2XML と称する XML 形式** [R1] のファイル群によって表現する方法は参考文書 [R2] に規定する。

**SIB2XML** では、上記の定義の各々を一つのファイルで表現する。**機能・情報定義**を作成するアプリケーションも**機能・情報定義**を利用するアプリケーションも **SIB2XML** をアクセスする。

**Memory Functional Object 定義**は、**Memory Functional Objects** について、それらの設計パラメータと**テレコマンド発行 SW** の機能に関する設定パラメータを定義する（4章参照）。

**Memory Functional Object 定義を、メモリ管理ファイル と称する XML 形式のファイル**によって表現する方法は参考文書 [R3] に規定する。

**Diagnostic Rule 定義**は、**Functional Object** が正常に動作しているかどうかを判定するものである **Diagnostic Rules** を定義する（6章参照）。なお、**自動監視ソフトウェア (Automated Monitoring Software, ATMOS)** を用いる事で、**Diagnostic Rule 定義**よりもはるかに複雑な診断ルールを定義する事も可能である（参考文書 [R4] 参照）。

The other subsidiary definitions consist of **Spacecraft Information Definition**, **MAP ID Definition**, and **Telemetry Packet Length Information Definition** (Sections 7.1, 7.2, and 7.3, respectively).

The information of **SIB2** can be shared in the ground system, using the **SIB2** system as described in Informative Reference [R5]. The information stored in **SIB2** might be input or edited using various formats (*e.g.*, table or diagram). Specific input or editing methods are defined separately for each individual application. Typical examples of specifications for specific methods are found in Informative References [R6] and [R7]. Many of the specific applications that deal with **SIB2** are described in Informative Reference [R5].

他の副次的な定義は、**宇宙情報定義**、**MAP ID 定義**、及び**テlemetryパケット長情報**からなる（それぞれ、7.1 項、7.2 項、及び 7.3 項）。

**SIB2** の情報は、地上系システムにおいて参考文書 [R5] が記すように **SIB2** システムを用いる事で共有する事ができる。**SIB2** に格納される情報は、様々な形式（例えば、表形式または図形式）を用いて入力または編集され得る。具体的な入力または編集方式は個々のアプリケーション毎に別途規定する。代表的なものは参考文書 [R6] 及び [R7] にある。**SIB2** を扱う具体的なアプリケーションの多くは参考文書 [R5] に記載されている。

## 2.5. Telemetry and Telecommand Value Processing // テレメトリ及びテレコマンド値の処理

### 2.5.1. (Sub-)Fields of the (Parameters and) Attribute Values field

#### (Parameters and) Attribute Values フィールドの（サブ）フィールド

In SIB2, one or more sub-fields (simply referred to as fields hereinafter) are defined in the (Parameters and) Attribute Values field of an SMCP Message.

In general, a field is either in fixed length or variable length. As a constraint, a field not located at the end of an SMCP Message is always in fixed length<sup>1</sup>.

In SIB2, the position where each field is located (*i.e.*, the octet position and bit position) is not explicitly specified but is implicitly specified through the specification of the order of the fields [Sections 3.2.5.4 Function Definition/Information Definition 1), 3.2.6.4 Function Definition/Information Definition 1), and 3.7.5.1 7)]. The way each field is handled is specified in Section 2.5.2.

SIB2 では、SMCP Message の (Parameters and) Attribute Values フィールドに一つ以上のサブフィールド（以下、単にフィールドと呼ぶ）を定義する。

一般に、各フィールドは固定長か可変長かの何れかである。制約として、SMCP Message<sup>2</sup>の末尾でないフィールドは、常に固定長である。

SIB2 では、各フィールドがどこに位置するか（つまり、octet 位置やビット位置）は明示的に定めず、フィールドの並び順を定める事により暗黙的に定める [3.2.5.4 項機能定義・情報定義 1)、3.2.6.4 項機能定義・情報定義 1)、及び 3.7.5.1 項 7)]. 各フィールドの扱いについては、2.5.2 項に定める。

---

<sup>1</sup> SIB2XML [R2] supports fields in variable length only for the ACTION Telecommand.

<sup>2</sup> SIB2XML [R2] は、可変長のフィールドを ACTION Telecommand に対してのみサポートしている。

### 2.5.2. Overview on Processing of Values of Fields // フィールドの値の処理の概観

The values handled in the processing of the values of a field of a telemetry or telecommand are categorized into [Binary Sequence Values](#), [Raw Data Values](#), [Engineering Values](#), and [String Character Values](#) (Sections 2.5.3, 2.5.4, 2.5.5, and 2.5.6, respectively).

Figure 2-1 shows the summary of the processing of values of fields specified in [SIB2](#), which is explained in the following subsections.

The processing consists of the following three steps.

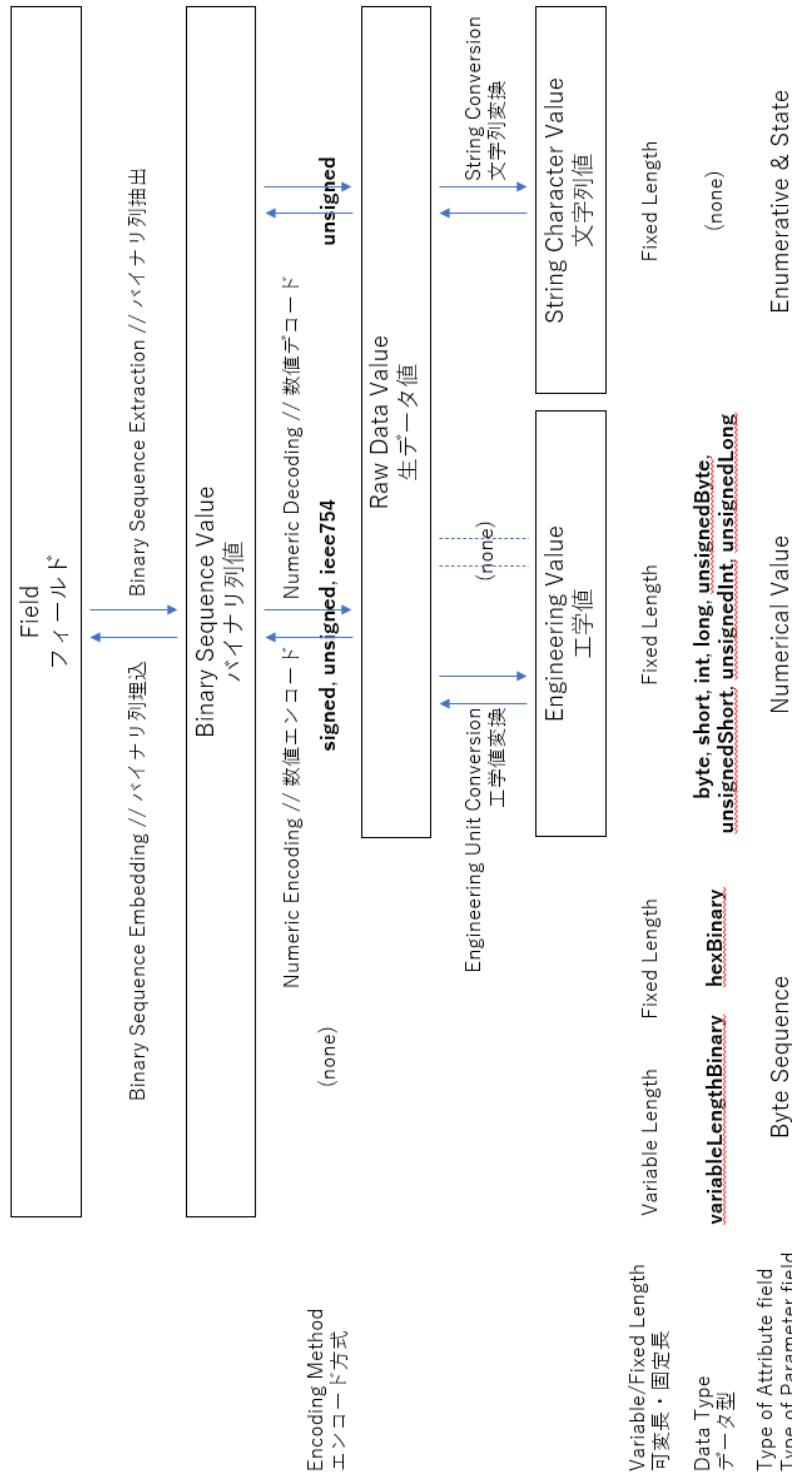
- [Binary Sequence Extraction/Embedding](#) (Section 2.5.3.4)
  - [Numerical Decoding/Encoding](#) (Section 2.5.4.3)
  - [Engineering Unit Conversion](#) (Section 2.5.5.2) or [String Conversion](#) (Section 2.5.6.2)
- Figure 2-1 に、本項で以降に説明する [SIB2](#) で定めるフィールドの値の処理のまとめを示す。
- この処理は、以下の三つのステップからなる。
- [バイナリ列抽出・埋込](#) (2.5.3.4 項)
  - [数値デコード・エンコード](#) (2.5.4.3 項)
  - [工学値変換](#) (2.5.5.2 項) または[文字列変換](#) (2.5.6.2 項)

テレメトリ・テレコマンドのフィールドの値の処理で扱う値は、[バイナリ列値](#)、[生データ値](#)、[工学値](#)、及び[文字列値](#)に分類される（それぞれ、2.5.3 項、2.5.4 項、2.5.5 項、及び 2.5.6 項）。

Figure 2-1 に、本項で以降に説明する [SIB2](#) で定めるフィールドの値の処理のまとめを示す。

この処理は、以下の三つのステップからなる。

- [バイナリ列抽出・埋込](#) (2.5.3.4 項)
- [数値デコード・エンコード](#) (2.5.4.3 項)
- [工学値変換](#) (2.5.5.2 項) または[文字列変換](#) (2.5.6.2 項)



**Figure 2-1: Summary of SIB2-Specified Processing of Values of Fields**

SIB2 に規定された、フィールドの値の処理のまとめ

### 2.5.3. Binary Sequence Value // バイナリ列値

#### 2.5.3.1. General // 一般

A sequence of binary values that constitutes the contents of a field is referred to as a [Binary Sequence Value](#) in this document.

本書では、各フィールドの中身を構成するものである、二進数値の並びを[バイナリ列値](#)と称する。

#### 2.5.3.2. Fixed Length or Variable Length // 固定長か可変長か

Each field is in either fixed length or variable length. Which of the two is appropriate for a field is specified according to Section 3.2.3.3 [Information Definition 2](#), Section 3.2.5.6 [Information Definition 1](#), and Section 3.2.6.5 [Information Definition 1](#).

各フィールドは固定長か可変長かの何れかである。各フィールドがその何れであるかを、3.2.3.3 項[情報定義 2](#)), 3.2.5.6 項[情報定義 1](#)), または 3.2.6.5 項[情報定義 1](#))に従って定める。

#### 2.5.3.3. Bit-Length

The bit length (if the field is in fixed length) or maximum bit length (if the field is in variable length) of a field, both of which is referred to as a [Bit-Length](#), is specified according to Section 3.2.3.3 [Information Definition 3](#), Section 3.2.5.6 [Information Definition 2](#)), and Section 3.2.6.5 [Information Definition 2](#)).

[Bit-Length](#) と称するものである、各フィールドのビット長（フィールドが固定長の場合）または最大ビット長（フィールドが可変長の場合）を、3.2.3.3 項[情報定義 3](#)), 3.2.5.6 項[情報定義 2](#)), または 3.2.6.5 項[情報定義 2](#))に従って定める。

The bit lengths of sub-fields for an [Attribute](#) are specified for each [Attribute](#). It is possible to specify different bit lengths depending on how the [Telemtries](#) and [Telecommands](#) are defined. Up to two lengths (called [tight](#) and [loose](#)) is allowed to be specified for an [Attribute](#) [3.2.3.3 [Information Definition 3](#)].

[Attribute](#) に対するフィールドのビット長は、[Attribute](#) 毎に定める。[Telemtries](#) 及び [Telecommands](#) の定義法に応じて異なるビット長を指定する事ができる。ある [Attribute](#) に対して、最大で二つの長さ ([tight](#) 及び [loose](#) 呼ばれる) を指定する事が許容される [3.2.3.3 [情報定義 3](#)]。

**tight** is for the definition method that often has many fields in one **SMCP Message** and specifies the bit length as compact as possible.

**loose** is for a definition method in which small number of fields is often transmitted at one time and is specified with priority given to ease of handling of field values on the spacecraft.

#### 2.5.3.4. Binary Sequence Embedding/Extraction // バイナリ列埋込・抽出

**Telecommand Issue SW** identifies the **octet** position, bit position, and bit length of each field according to its order and **Bit-Length** (see Sections 2.5.1 and 2.5.3.3, respectively), and embeds the **Binary Sequence Value** into the field of a **Telecommand Message** to be transmitted. This operation is referred to as

##### Binary Sequence Embedding.

The **Data Conversion Module** performs the operation in the reverse order of the above-described one on a received **Telemetry Message**. This operation is referred to as Binary Sequence Extraction.

**tight** は一つの **SMCP Message** に多くのフィールドを持つ事が多い定義法のためのものであり、可能な限りコンパクトなビット長を指定する。

**loose** は、1 度に伝送するフィールドの数が少ない事が多い定義法のためのものであり、宇宙機でのフィールドの値の扱いやすさを優先して指定する。

テレコマンド発行 SW は、各フィールドの並び順と **Bit-Length** (それぞれ、2.5.1 項及び 2.5.3.3 項参照) に従ってフィールドの **octet** 位置、ビット位置、ビット長を特定し、送信する **Telecommand Message** の各フィールドに バイナリ列値 を埋め込む。この操作を バイナリ列埋込 と称する。

データ変換モジュールは、受信した **Telemetry Message** に対して、上記と逆の流れの操作を実施する。この操作を バイナリ列抽出 と称する。

## 2.5.4. Raw Data Value // 生データ値

### 2.5.4.1. General // 一般

A **Binary Sequence Value** might be a representation of a numeric value, in which case the numeric value is referred to as a Raw Data Value in this document.

**バイナリ列値**は、数値を表現する値である事がある。その場合、その数値を、本書では生データ値と称する。

### 2.5.4.2. Encoding Method // エンコーディング方式

If a **Binary Sequence Value** is a representation of a numeric value, an Encoding Method (Section 3.1.3) is specified for the **Binary Sequence Value**. The **Encoding Method** of a field is one of the unsigned integer (**unsigned**), signed integer (two's complement) (**signed**), and IEEE754-format floating-point number (**ieee754**). For an integer (**unsigned** or **signed**), the bit length of a field must be between 1 bit and 64 bits. For a floating-point number (**ieee754**), the bit length of a field must be 32 bits (single precision) or 64 bits (double precision).

**バイナリ列値**が数値の表現である場合、その**バイナリ列値にエンコーディング方式** (3.1.3 項) を定める。あるフィールドの**エンコーディング方式**は、符号なし整数 (**unsigned**)、符号あり整数 (二の補数) (**signed**)、IEEE754 形式の浮動小数点数 (**ieee754**) の何れかである。整数 (**unsigned** または **signed**) の場合、フィールドのビット長は 1 ビットから 64 ビットである必要がある。浮動小数点数 (**ieee754**) の場合、フィールドのビット長は 32 ビット (単精度) か 64 ビット (倍精度) である必要がある。

### 2.5.4.3. Numerical Encoding/Decoding // 数値エンコード・デコード

The operation to obtain a **Binary Sequence Value** in the specified **Encoding Method** from a **Raw Data Value** is referred to as **Numerical Encoding**.

Telecommand Issue SW has a function to perform Numerical Encoding according to the Encoding Method [Sections 3.2.3.3 Information Definition 3) and 3.2.5.6 Information Definition 3)].

The operation to obtain a **Raw Data Value** from a **Binary Sequence Value** in the specified **Encoding Method** is referred to as **Numerical Decoding**.

The **Data Conversion Module** has a function to perform Numerical Decoding according to the Encoding Method [Sections 3.2.3.3 Information Definition 3) and 3.2.6.5 Information Definition 3)].

生データ値から、指定したエンコーディング方式のバイナリ列値を得る操作を**数値エンコード**と称する。

テレコマンド発行 SW は、エンコーディング方式 [3.2.3.3 項情報定義 3)及び 3.2.5.6 項情報定義 3)] に従って、数値エンコードを実施する機能を持つ。

指定したエンコーディング方式のバイナリ列値から、生データ値を得る操作を**数値デコード**と称する。

データ変換モジュールは、エンコーディング方式 [3.2.3.3 項情報定義 3)及び 3.2.6.5 項情報定義 3)] に従って、数値デコードを実施する機能を持つ。

## 2.5.5. Engineering Value // 工学値

### 2.5.5.1. General // 一般

In this document, a numeric value that expresses a physical quantity is referred to as an Engineering Value. Each of Numerical Value Attributes, Numerical Value Parameters, and Pseudo Attributes (Section 2.5.8) has an Engineering Value.

If a Raw Data Value itself does not represent a physical quantity, it can be associated with an Engineering Value by specifying an Engineering Unit Conversion described in Section 2.5.5.2. If no Engineering Unit Conversion is specified for a field, the Raw Data Value and Engineering Value of the field are identical.

### 2.5.5.2. Engineering Unit Conversion // 工学値変換

The Engineering Unit Conversion specifies either the conversion between a Raw Data Value and an Engineering Value or the formula to calculate an Engineering Value from Engineering Values. One Engineering Unit Conversion defines one of an Encode Engineering Unit Conversion (Section 2.5.5.3) for telecommand values, a Decode Engineering Unit Conversion (Section 2.5.5.4) for telemetry values, and the formula to calculate a Pseudo Attribute (Section 2.5.8).

[Note] An Engineering Value is the value of one of a Numerical Value Attribute, Numerical Value Parameter, and Pseudo Attribute.

本書では、物理量を表す数値を工学値と称する。Numerical Value Attributes, Numerical Value Parameters, 及び Pseudo Attributes (2.5.8 項) の各々は工学値を持つ。

生データ値は、それ自身が物理量を表していない場合、2.5.5.2 項に記す工学値変換を定め、工学値と対応させる事ができる。あるフィールドに、工学値変換を定めない場合、そのフィールドの生データ値と工学値は同一である。

工学値変換は、生データ値と工学値の間の変換、または、(一つまたは複数の)工学値から一つの工学値への計算式の何れかを定めるものである。一つの工学値変換は、テレコマンド値に対する Encode 工学値変換 (2.5.5.3 項) か、テlemetry 値に対する Decode 工学値変換 (2.5.5.4 項) か、Pseudo Attribute (2.5.8 項) の計算式 か、の何れかを定めるものである。

[注] 工学値は、Numerical Value Attribute, Numerical Value Parameter, または Pseudo Attribute の何れかの値である。

### 2.5.5.3. Encode Engineering Unit Conversion // Encode 工学値変換

Telecommand Issue SW has a function to perform Engineering Unit Conversions referred to as Encode Engineering Unit Conversions.

In an Encode Engineering Unit Conversion, the Engineering Value of a Numerical Value Attribute or Numerical Value Parameter is converted to its Raw Data Value according to Sections 3.2.3.5 Information Definition 2) and 3.2.5.8 Information Definition 1).

テレコマンド発行 SW は、Encode 工学値変換と称する工学値変換を実施する機能を持つ。

各 Encode 工学値変換では、3.2.3.5 項情報定義 2) 及び 3.2.5.8 項情報定義 1) に従って、Numerical Value Attribute または Numerical Value Parameter の工学値が、生データ値へ変換される。

### 2.5.5.4. Decode Engineering Unit Conversion // Decode 工学値変換

The Data Conversion Module has a function to perform Engineering Unit Conversions referred to as Decode Engineering Unit Conversions.

In an Decode Engineering Unit Conversion, the Raw Data Value of a Numerical Value Attribute or Numerical Value Parameter is converted to its Engineering Value according to Sections 3.2.3.5 Information Definition 1) and 3.2.6.7 Information Definition 1).

データ変換モジュールは、Decode 工学値変換と称する工学値変換を実施する機能を持つ。

各 Decode 工学値変換では、3.2.3.5 項情報定義 1) 及び 3.2.6.7 項情報定義 1) に従って、Numerical Value Attribute または Numerical Value Parameter の生データ値が、工学値へ変換される。

## 2.5.6. String Character Value // 文字列値

### 2.5.6.1. General // 一般

In this document, the value of string characters is referred to as a String Character Value. Each of Enumerative Attributes, Enumerative Parameters, and State Attributes has a String Character Value.

The Raw Data Value is converted to/from a String Character Value using the String Conversion described in Section 2.5.6.2.

本書では文字列の値を文字列値と称する。  
Enumerative Attributes, Enumerative Parameters 及び State Attributes の各々は文字列値を持つ。

生データ値は、2.5.6.2 項に記す文字列変換を用いて、文字列値と変換される。

### 2.5.6.2. String Conversion // 文字列変換

A String conversion converts either a Raw Data Value to a String Character Value (called Decode Conversion) or a String Character Value to a Raw Data Value (called Encode Conversion). The definition of a String Conversion has pairs of a Raw Data Value and a String Character Value and is used in both the Encode Conversions and Decode Conversions. The uniqueness in the conversion is guaranteed for the String conversion.

Telecommand Issue SW has a function to perform the Encode Conversion of the telecommand value of the Enumerative Attribute according to the Enumeration Definition (*i.e.*, the correspondence between a Raw Data Value and an Enumerative Name specified in Section 3.9.2.2 Information Definition 1) and Function Definition 1), respectively).

The Data Conversion Module has functions to perform the Decode Conversion of the telemetry value of the Enumerative Attribute according to the Enumeration Definition and to perform the Decode Conversion of the telemetry value of the State Attribute according to the State Machine Definition (*i.e.*, the correspondence between a Raw Data Value and a State name specified in Section 3.3.2.2 Information Definition 1) and Function Definition 1), respectively).

文字列変換は、生データ値を文字列値に変換するもの(Decode 変換と呼ぶ)、または、文字列値を生データ値に変換するもの(Encode 変換と呼ぶ)の何れかである。一つの文字列変換の定義は、生データ値と文字列値の対応の組を持ち、Encode 変換とDecode 変換の双方で用いられる。文字列変換では、変換の一意性が保証される。

テレコマンド発行 SW は、Enumeration 定義(つまり、3.9.2.2 項情報定義 1)が定める生データ値と機能定義 1)が定める Enumerative Name の対応)に従って Enumerative Attribute のテレコマンド値の Encode 変換を実施する機能を持つ。

データ変換モジュールは、Enumeration 定義に従って Enumerative Attribute のテレメトリ値の Decode 変換を実施する機能、並びに、State Machine 定義(つまり、3.3.2.2 項情報定義 1)が定める生データ値と機能定義 1)が定める State 名の対応)に従って State Attribute のテレメトリ値の Decode 変換を実施する機能を持つ。

## 2.5.7. Type of Attribute or Parameter fields // 属性またはパラメーターフィールドの種別

### 2.5.7.1. General // 一般

There are four types of **Attributes** and **Parameters**: **Binary Sequence**, **Numerical Value**, **Enumerative**, and **State** (for the **Attribute** only). The length of the field, encoding/decoding of values, and handling of them depend on the type as described in the following.

### 2.5.7.2. Data Type // データ型

The **Data Type** (Section 3.1.2) is a concept to indicate whether the length of the field is variable or fixed, whether the type of the **Attribute/Parameter** is a **Numerical Value** or **Binary Sequence**, and how its **Engineering Value**, if it is a **Numerical Value**, is handled in computers (both aboard the spacecraft and on the ground).

[Note 1] This document does not specify how the **Data Type** of a **Numerical Value Attribute/Parameter** is interpreted and the interpretation method is specified in each implementation that handles **SIB2**.

[Note 2] No **Data Type** is specified for a field that holds a value in **Enumeration** or **State**.

**Attributes** と **Parameters** には、**Binary Sequence**, **Numerical Value**, **Enumerative**, 及び **State** (**Attribute** についてのみ) の四つの種別がある。以下に示すように、フィールドの長さ、値のエンコード、デコード及び、扱いはこの種別に依存する。

**データ型** (3.1.2 項) は、フィールドが可変長か固定長か、**Attribute/Parameter** の種別が **Numerical Value** か **Binary Sequence** であるか、また **Numerical** の場合は計算機（宇宙機搭載と地上の双方）で **工学値** を扱う方法を示す概念である。

[注 1] 本書は **Numerical Value Attribute/Parameter** の **データ型** の解釈方法は規定しておらず、解釈方法は **SIB2** を扱う処理系毎に定める。

[注 2] **Enumerative** または **State** の値を保持するフィールドには、**データ型** を定めない。

### 2.5.7.3. Binary Sequence Attribute/Parameter field

The length of a field that holds the value of a **Binary Sequence Attribute/Parameter** is either fixed or variable.

The **Data Type** of the fixed-length or variable-length field is **hexBinary** or **variableLengthBinary**, respectively.

No **Encoding Method** is specified for the field in **SIB2**. Thus, the **Binary Sequence Value** of the field presents neither a **Raw Data Value** nor an **Engineering Value** in the model of **SIB2**.

[Note] A **Binary Sequence Attribute/Parameter** is used to represent a value of a more complicated data type (such as an array or structure type)

### 2.5.7.4. Numerical Value Attribute/Parameter field

A field that holds the value of a **Numerical Value Attribute/Parameter** has a fixed length between 1 bit and 64 bits.

The **Encoding Method** of the field is one of “**unsigned**”, “**signed**”, and “**ieee754**”.

The **Data Type** of the field is one of “**byte**”, “**unsignedByte**”, “**short**”, “**unsignedShort**”, “**int**”, “**unsignedInt**”, “**long**”, “**unsignedLong**”, “**float**”, and “**double**” (see Section 3.1.2.3).

An **Encode Engineering Unit Conversion** and/or **Decode Engineering Unit Conversion** might be specified for the field.

**Binary Sequence Attribute/Parameter** の値を保持するフィールドは、固定長または可変長の何れかである。

**データ型**は、固定長フィールドか可変長フィールドかに応じ、それぞれ、**hexBinary** または **variableLengthBinary** である。

このフィールドには、**SIB2** では **エンコーディング方式**を定めない。そこで、**SIB2** のモデルでは、このフィールドの **バイナリ列値**は、**生データ値**も**工学値**も表現しない。

[注] **Binary Sequence Attribute/Parameter** は、より複雑なデータ型（配列型や構造型等）の値の表現に用いる。

**Numerical Value Attribute/Parameter** の値を保持するフィールドは、1 ビットから 64 ビットまでの固定長である。

このフィールドの **エンコーディング方式**は “**unsigned**”, “**signed**”, または “**ieee754**” の何れかである。

このフィールドの **データ型**は、“**byte**”, “**unsignedByte**”, “**short**”, “**unsignedShort**”, “**int**”, “**unsignedInt**”, “**long**”, “**unsignedLong**”, “**float**”, または “**double**” の何れかである (3.1.2.3 項参照)。

さらに、**Encode 工学値変換**と **Decode 工学値変換**の一方または双方を定める事ができる。

### 2.5.7.5. **Enumerative Attributes/Parameter** field and **State Attribute** field

A field that holds the value of an **Enumeration** or **State Attribute/Parameter** has a fixed length between 1 bit and 32 bits.

The **Encoding Method** of the field is **unsigned**, i.e., the **Raw Data Value** of a field that holds the value of an **Enumeration** or **State Attribute/Parameter** is a positive integer.

No **Data Type** is specified for the field.

A **String Conversion** is specified for the field.

**Enumerative** または **State Attribute/Parameter** の値を保持するフィールドは、1 ビットから 32 ビットまでの固定長である。

このフィールドの **エンコーディング方式** は **unsigned** である。つまり、**Enumerative** または **State Attribute/Parameter** の値を保持するフィールドの **生データ値** は正の整数である。

このフィールドには **データ型** は定めない。

このフィールドには **文字列変換** を定める。

### 2.5.8. Pseudo Attribute

The value of a Pseudo Attribute is an Engineering Value that is calculated from other Engineering Values. A Pseudo Attribute is handled in the same way as an Attribute in application programs that handle Engineering Values.

The Data Conversion Module has a function to calculate the value of the Pseudo Attribute according to the information specified in Section 3.2.4.2 Information Definition 3).

Pseudo Attribute の値は、他の 工学値 から計算される 工学値 である。 Pseudo Attribute は、工学値 を扱う アプリケーション プログラムにおいて Attribute と同様に扱われる。

データ変換モジュール は、3.2.4.2 項 情報定義 3 に示す情報に従って Pseudo Attribute の値を計算する機能を持つ。

## 2.6. Functions of Telecommand Issue SW and Telemetry Monitor SW

### テレコマンド発行 SW 及びテlemetry Monitor SW の機能

#### 2.6.1. General // 一般

In this section, the functions of Telecommand Issue SW and Telemetry Monitor SW are specified. Notably, the functions include those of a Data Conversion Module, which are common functions of the two SWs.

本項には、テレコマンド発行 SW とテlemetry Monitor SW の機能、特にこれら二つの SWs の共通機能であるデータ変換モジュールの機能を含めて記す。

### 2.6.2. Issuance of Telecommands // Telecommands の発行

Telecommand Issue SW has a function to issue Telecommands.

In the issuance, Telecommand Issue SW informs the human operator of the Criticality Level of the Telecommand if needed and checks whether the issuer is allowed to issue the Telecommand or not according to the information specified in Section 3.2.5.4 Function Definition 4) and

Information Definition 2).

Telecommand Issue SW also checks whether the values of the parameters of Telecommands to be issued satisfy the constraints or not according to the Valid Range described in Sections 3.2.3.5 Function Definition 1) and 3.2.5.8 Function Definition 1).

テレコマンド発行 SW は、Telecommands を発行する機能を持つ。

発行に際して、テレコマンド発行 SW は、3.2.5.4 項機能定義 4) 及び情報定義 2) に定める情報に従って、Telecommand の Criticality Level を必要に応じ操作者に知らせると共に発行主体がその Telecommand を発行する事を許容するか否かをチェックする。

また、テレコマンド発行 SW は、発行する Telecommands のパラメータの値が制約を満たすか否かも、3.2.3.5 項機能定義 1) 及び 3.2.5.8 項機能定義 1) に記す Valid Range に従ってチェックする。

### 2.6.3. Precheck of Telecommands

Telecommand Issue SW has a function to check whether the issuance of a Telecommand is allowed or not before the issuance according to the information of the Effective-Condition or Begin State name described in Section 3.2.2.1 Function Definition 2), Section 3.2.5.4 Function Definition 2), and Section 3.3.2.3 Function Definition 2). The function is called the Precheck.

テレコマンド発行 SW は、Telecommand を発行してよいか否か、3.2.2.1 項機能定義 2), 3.2.5.4 項機能定義 2), 及び 3.3.2.3 項機能定義 2) に記す情報、有効条件 または Begin State 名、に従って、発行に先立ちチェックする機能を持つ。この機能を Precheck と呼ぶ。

## 2.6.4. Collations of Telecommands // Telecommands の照合

### 2.6.4.1. General // 一般

Telecommand Issue SW has a function to perform Collations for Telecommands. When Telecommand Issue SW issues a Telecommand, it starts the Collation of the Telecommand. There are four types of Collations of Telecommands: Echo Collation (Section 2.6.4.2), Send Verification (Section 2.6.4.3), Acknowledge Collation (Ack Collation); Section 2.6.4.4), and Success Verification (Section 2.6.4.5).

Upon an issuance of a telecommand, Telecommand Issue SW always performs an Echo Verification. Whether Telecommand Issue SW performs zero or more of Send Verifications, Ack Collations, and Success Verifications are specified in the setting of Telecommand Issue SW.

When Telecommand Issue SW issues a Telecommand, whether Telecommand Issue SW waits or not for the completion of the Collations of the previous Telecommands before the issuance of the Telecommand is specified in the setting of Telecommand Issue SW. If Telecommand Issue SW does not wait for the completion of the Collations of the previous Telecommands, Telecommand Issue SW performs the Collations of multiple Telecommands in parallel.

テレコマンド発行 SW は、Telecommands の照合を実施する機能を持つ。テレコマンド発行 SW は、Telecommand を発行すると、その Telecommand に対する照合を開始する。Telecommands の照合には、エコー照合 (2.6.4.2 項), センドベリファイ (2.6.4.3 項), Acknowledge 照合 (Ack 照合); 2.6.4.4 項), サクセスベリファイ (2.6.4.5 項) の四種類が存在する。

テレコマンド発行 SW は、あるテレコマンドの発行において、エコー照合は常に実施する。テレコマンド発行 SW が、センドベリファイ、Ack 照合、サクセスベリファイの各々を実施するかは、テレコマンド発行 SW の設定により指定する。

テレコマンド発行 SW がある Telecommand を発行する際、その発行前にテレコマンド発行 SW がそれ以前の Telecommands の照合が完了するのを待つか待たないかは、テレコマンド発行 SW の設定により指定する。テレコマンド発行 SW は、それ以前の Telecommands 照合の完了を待たない場合、複数の Telecommands の照合を同時並行で実施する。

#### 2.6.4.2. Echo Collation // エコー照合

In the Echo Collation of a Telecommand, Telecommand Issue SW verifies that the Telecommand has been sent from the ground station, using the response from the ground station (see Section 3.3 in [A3]).

ある Telecommand のエコー照合では、テレコマンド発行 SW は、地上局から Telecommand が送信された事を、地上局からの応答 ([A1] 3.3 項参照) を用いて検証する。

#### 2.6.4.3. Send Verification // センドベリファイ

In the Send Verification of a Telecommand, Telecommand Issue SW verifies, according to the Communications Operation Procedure-1 (COP-1) [A7], by using Communications Link Control Words (CLCWs) received from the spacecraft, that the spacecraft has received the Telecommand.

ある Telecommand のセンドベリファイでは、テレコマンド発行 SW は、宇宙機がその Telecommand を受信した事を、Communications Operation Procedure-1 (COP-1) [A7] に従い、同宇宙機から受信する Communications Link Control Words (CLCWs) を用いて検証する。

#### 2.6.4.4. Acknowledge Collation (Ack Verification) // Acknowledge 照合(Ack 照合)

In the Acknowledge Verification (Ack Verification) of a Telecommand, Telecommand Issue SW verifies, according to the SMCP [A3], by using ACK Telemetry received from the Functional Object, that the Functional Object to which the Telecommand is sent to has received the Telecommand<sup>3</sup>.

ある Telecommand のAcknowledge 照合 (Ack 照合)では、テレコマンド発行 SW は、ある Telecommand の送信先の Functional Object が、その Telecommand を受信した事を、SMCP [A3] に従い、同 Functional Object から受信する ACK Telemetry を用いて検証する<sup>4</sup>。

---

<sup>3</sup> The Ack Verification is not in the current implementation as of March 2023.

<sup>4</sup> 2023 年 3 月現在の実装で Ack 照合は未実装である。

### 2.6.4.5. Success Verification // サクセスベリファイ

In the Success Verification of a **Telecommand**, **Telecommand Issue SW** verifies that the **Functional Object** has successfully executed the **Telecommand**, by checking values of the **Telemetries** sent from the **Functional Object** (hereafter relevant **Telemetries**) [Section 3.2.5.3 **Function Definition** 1) **ParameterValueSetting Attribute name**, Section 3.2.5.4 **Functionalal Definition** 3) **Attribute Change Rule name**, and Section 3.3.2.3 **Function Definition** 3) **End State name**].

**Telecommand Issue SW** waits for the duration specified [Section 3.4.2.2 **Information Definition** 1) **Wait Time** and Section 3.3.2.3 **Function Definition** 4) **Maximum Allowable Transition Time** and 5) **Minimum Allowable Transition Timel**] from the completion of the **Send Verification** if **Send Verification** is performed or from the completion of the **Echo Collation** otherwise .

After that, **Telecommand Issue SW** receives relevant **Telemetries** up to a specified number of times [Section 3.4.2.2 **Information Definition** 2) **Maximum Receive Times**] and checks whether the **Engineering Value** is an expected value.

**Telecommand Issue SW** declares success when the expected value is obtained and terminates the **Success Verification**.

ある **Telecommand のサクセスベリファイ**では、**テレコマンド発行 SW**は、**Functional Object**がその **Telecommand**を成功裏に実行した事を、同 **Functional Object**が送付する **Telemetries** (以降、関連 **Telemetries**) の値により検証する [3.2.5.3 項**機能定義** 1) **パラメータ 値設定 Attribute 名**、3.2.5.4 項**機能定義** 3) **Attribute Change Rule name**, 及び 3.3.2.3 項**機能定義** 3) **End State 名**]。

**テレコマンド発行 SW**は、**センドベリファイ**を実施する場合は**センドベリファイ**完了時から、さもなければ**エコー照合**完了時から、指定した時間 [3.4.2.2 項**情報定義** 1) **待ち時間**及び 3.3.2.3 項**機能定義** 4) **Maximum Allowable Transition Time** 及び 5) **Minimum Allowable Transition Time**] 経過するのを待つ。

その後、**テレコマンド発行 SW**は、関連 **Telemetries**を指定回数 [3.4.2.2 項**情報定義** 2) **最大再受信回数**]まで受信し、**その工学値**が期待した値であるかチェックする。

**テレコマンド発行 SW**は、期待した値が得られた時点で成功と判断し、**サクセスベリファイ**を終了する。

### 2.6.5. Limit Check

The Data Conversion Module has a function called the Limit Check, which checks whether the Engineering Values of Numerical Value Attributes and Pseudo Attributes are within the ranges defined in Section 3.2.3.5 Function Definition 2) and Section 3.2.4.2 Function Definition 5), respectively.

データ変換モジュールは、Limit Check と呼ばれる機能を持つ。この機能は、Numerical Value Attributes 及び Pseudo Attributes の工学値が、それぞれ、3.2.3.5 項機能定義 2) 及び 3.2.4.2 項機能定義 5)で定義する範囲にあるかチェックするものである。

## 2.6.6. Character Display // 文字表示

### 2.6.6.1. General // 一般

Telemetry Monitor SW has a function of Character Display. In this function, Telemetry Monitor SW shows 1) Binary Sequence Values, Raw Data Values, String Character Values, and Engineering Values of Attributes and 2) the Engineering Values of Pseudo Attributes according to the definition of the display.

### 2.6.6.2. Engineering Value // 工学値

If the Engineering Value of a Numerical Value Attribute or Pseudo Attribute is displayed with characters, Telemetry Monitor SW sets the color of the characters as follows:

- (1) If the Engineering Value is outside the range of its Action Limits (*i.e.*, if the Critical Level is Action) [Section 3.5.2.2 Function Definition 2)], Telemetry Monitor SW sets the color of the characters to red.
- (2) If the Engineering Value is within the Action Limits and is outside the Caution Limits (*i.e.*, if the Critical Level is Caution) [Section 3.5.2.2 Function Definition 1)], Telemetry Monitor SW sets the color of the characters to yellow.
- (3) Otherwise, the Engineering Value is within the Action Limits and is within the Caution Limits (*i.e.*, the Critical Level is Normal). In this case, Telemetry Monitor SW sets the color of the characters to white.

テレメトリ監視 SW は、文字表示の機能を持つ。この機能は、1) Attributes のバイナリ列値、生データ値、文字列値、及び工学値、並びに、2) Pseudo Attributes の工学値を画面の定義に基づき表示する。

ある Numerical Value Attribute または Pseudo Attribute の工学値が文字で表示される場合、テレメトリ監視 SW は文字色を以下のように設定する。

- (1) 工学値が Action Limits の範囲外（つまり、Critical Level が Action）[3.5.2.2 項機能定義 2)] である場合、テレメトリ監視 SW は文字色を赤に設定する。
- (2) 工学値が Action Limits の範囲内であり、かつ、Caution Limits の範囲外である（つまり、Critical Level が Caution）[3.5.2.2 項機能定義 1)] 場合、テレメトリ監視 SW は文字色を黄に設定する。
- (3) それ以外の場合、工学値が Action Limits の範囲内、かつ、Caution Limits の範囲内である（つまり、Critical Level が Normal）。この場合、テレメトリ監視 SW は文字色を白に設定する。

### 2.6.6.3. String Character Value // 文字列値

If Telemetry Monitor SW displays the character string obtained through the String Conversion for an Enumerative Attribute or State Attribute, the color of the character string is set to the color specified according to its Criticality Level [Section 3.3.2.2 Function Definition 3) or Section 3.9.2.2 Function Definition 2), respectively].

テレメトリ監視 SW は、ある Enumerative Attribute または State Attribute に対して、文字列変換を通じて得られた文字列を表示する場合、その Criticality Level [それぞれ、3.3.2.2 項機能定義 3)または 3.9.2.2 項機能定義 2)] に従つて指定した色に設定する。

## 2.6.7. Management and Operations of the Contents of Memories // メモリの内容の管理と操作

Telecommand Issue SW has a function to manage/operate the contents of Memory Functional Objects (hereafter referred to as Memories in this section). This function is called Memory Management.

Telecommand Issue SW uploads (LOAD) planned contents of a Memory from the ground to the Memory on a spacecraft and downlinks (DUMP) contents of the Memory from the spacecraft to the ground. If a Memory is a target of the Memory Management [Section 4.2.5)], Telecommand Issue SW stores the downlinked contents as the expected value of the Memory.

There are two methods to upload contents of a Memory: 1) upload the entire planned contents (without Optimization), 2) upload only the parts where the planned contents differ from the contents of the Memory (with Optimization). The former is used when the contents of Memory are unknown. The latter is used to minimize upload time when the contents of the Memory are known.

The commands (to Telecommand Issue SW) that operate a Memory are classified into four types listed in Table 2-1, which are the combinations of 1) write or read, and 2) a telecommand or a CISW Control Command (Section 2.3).

A telecommand targets a portion of the address range of the Memory. Here, read might target the entire Memory and write only targets a

テレコマンド発行 SW は Memory Functional Objects (以下、本項では Memories と称する) の内容を管理・操作する機能を持つ。この機能は、メモリ管理と呼ばれる。

テレコマンド発行 SW は、ある Memory に対して、計画した内容を地上から宇宙機の Memory へアップロード (LOAD) する。また、その Memory の内容を、宇宙機から地上へダウンリンク (DUMP) する。ある Memory を メモリ管理 の対象とした [4.2 項 5)] 場合、テレコマンド発行 SW は、Memory の期待値として、ダウンリンクされた内容を蓄積する。

Memory の内容をアップロードする方法は二つある: 1) 計画した内容全体のアップロード (最適化なし)、2) 計画した内容が Memory の内容と異なる部分のみのアップロード (最適化あり)。前者は Memory の内容が不明な場合に用いる。後者は Memory の内容が既知な場合、アップロード時間を最短にするのに用いる。

Memory を操作する(テレコマンド発行 SW への) コマンドは、1) 書き込みか読み出しか、2) テレコマンドか、CISW制御コマンド (2.3 項) かの何れかである。結果、それらの組み合わせに応じて、Table 2-1 に記す四種類に分類される。

テレコマンドはその Memory のあるアドレス範囲を対象とする。ここで、読み出しは Memory 全体も対象としうるが、書き込みは Memory の

limited portion of the **Memory**.

A **CISW Control Command** always targets the entire address range of the **Memory**.

限られた部分のみを対象とする。

**CISW 制御コマンド**は常にその **Memory** の全アドレス範囲を対象とする。

**Table 2-1: Classification of the commands that operate the contents of a Memory**  
**Memory の内容を操作するコマンドの分類**

	Telecommand ( <b>SMCP Telecommand</b> ) // テレコマンド ( <b>SMCP Telecommand</b> )	<b>CISW Control Command</b> // <b>CISW 制御コマンド</b>
targets // 対象	a portion of the address range of the <b>Memory</b> // <b>Memory</b> のあるアドレス範囲	the entire address range of the <b>Memory</b> // <b>Memory</b> の全アドレス範囲
Write // 書き込み	<b>WRITE Command</b> (= <b>MEMORY LOAD Telecommand</b> )	<b>LOAD Command</b>
Read // 読み出し	<b>READ Command</b> (= <b>MEMORY DUMP Telecommand</b> )	<b>DUMP Command</b>

The **WRITE Command** is a command to write data into a specific address range of a **Memory**, which corresponds to the **MEMORY LOAD Telecommand** in the **SMCP**.

The **LOAD Command** is a **CISW Control Command** that commands **Telecommand Issue SW** to update the contents in the entire address range of a **Memory**.

The **READ Command** is a command to read data from a specific address range of a **Memory**, which corresponds to the **MEMORY DUMP Telecommand** in the **SMCP**.

The **DUMP Command** is a **CISW Control Command** that commands **Telecommand Issue SW** to read the contents in the entire address range of a **Memory**.

[Note] For a **DUMP Command**, the current **Telecommand Issue SW** only has a function to issue one **MEMORY DUMP Telecommand**.

**WRITE Command** は、**Memory** の特定のアドレス範囲にデータを書き込むためのコマンドである。**SMCP** の **MEMORY LOAD Telecommand** に対応する。

**LOAD Command** は、ある **Memory** の全アドレス範囲の内容の更新を、**テレコマンド発行 SW** へ指示する **CISW 制御コマンド**である

**READ Command** は、**Memory** の特定のアドレス範囲からデータを読み出すためのコマンドである。 **SMCP** の **MEMORY DUMP Telecommand** に対応する。

**DUMP Command** は、ある **Memory** の全アドレス範囲の内容の読み出しを、**テレコマンド発行 SW** へ指示する **CISW 制御コマンド**である。

[注] 現在の **テレコマンド発行 SW** は、**DUMP Commands** に対し、一つの **MEMORY DUMP Telecommand** を発行する機能のみを持つ。

[Note] The wordings of 'LOAD' and 'DUMP' [注] ここで、SMCP Telecommands と CISW 制 indicate different meanings between the 御コマンドで LOAD, DUMP という表現が意味 SMCP Telecommands and the CISW Control するものが異なることに注意。 Commands.

WRITE Command = MEMORY LOAD Telecommand ≠ LOAD Command  
READ Command = MEMORY DUMP Telecommand ≠ DUMP Command

### 3. Function/Information Definitions // 機能・情報定義

#### 3.1. Overall Structure and Common Definitions // 全体構成及び共通定義

##### 3.1.1. General // 一般

In the **Function/Information Definition**, the items of the **Function Definition** and **Information Definition** and their multiplicities are specified in the following notation in each definition. Here, if the multiplicity is 1, the notation of multiplicity “(1)” is omitted. If the permitted multiplicity is 0 or 1, *i.e.*, “(0–1)”, it is written as “(optional)”.

###### [Function Definition]

###### 1) **Function Definition** item (multiplicity)

::= Definition of the item

Constraints

**機能・情報定義**では、定義毎に以下の表記で**機能定義**及び**情報定義**の各項目とその多重度を規定する。ただし多重度が 1 の場合、多重度の表記“(1)”は省略する。また、許される多重度が 0 または 1 すなわち “(0 ~ 1)” の場合は “(optional)”と表記する。

###### [機能定義]

###### 1) **機能定義**項目（多重度）

::= 項目の定義

制約事項

###### [Information Definition]

###### 1) **Information Definition** item (multiplicity)

::= Definition of the item

Constraints

###### [情報定義]

###### 1) **情報定義**項目（多重度）

::= 項目の定義

制約事項

The **Function/Information Definition** shall consist of a **Functional Object Definition**, **State Machine Definition**, **Attribute Change Rule Definition**, **Attribute Limit Definition**, **Condition and Event Definition**, **Attribute Sequence Definition**, **Conversion Definition**, and **Enumeration Definition** (Sections 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9, respectively).

**機能・情報定義**は、**Functional Object 定義**、**State Machine 定義**、**Attribute Change Rule 定義**、**Attribute Limit 定義**、**Condition and Event 定義**、**Attribute Sequence 定義**、**Conversion 定義**、及び**Enumeration 定義**（それぞれ、3.2 項、3.3 項、3.4 項、3.5 項、3.6 項、3.7 項、3.8 項、及び 3.9 項）からなること。

In this section, the definition items commonly used in all or many of the definitions are specified.

本項では、これらの定義の全てまたは多くで共通に使用する定義項目について規定する。

### 3.1.2. Data Types and Properties of Fields // データ型とフィールドの特性

#### 3.1.2.1. General // 一般

This section specifies field characteristics (fixed length or variable length described in Section 2.5.3.2 and bit length described in Section 2.5.3.3) and Data Types. Note that the Data Types are specified only for the Attribute/Parameter whose type is Numerical Value or Binary Sequence.

The Data Type of the Binary Sequence Attribute/Parameter is a Binary Sequence Data Type, which specifies whether the lengths of the corresponding fields are variable or fixed. See Section 3.1.2.2 for the Binary Sequence Data Types and properties of the corresponding fields.

The Data Type of the Numerical Value Attribute/Parameter is a Numerical Value Data Type, which specifies how its Engineering Value (Section 2.5.5) are handled with computers (both aboard the spacecraft and on the ground). See Section 3.1.2.3 for the Numerical Value Data Types and properties of corresponding fields.

[Note] How the Data Type of a Numerical Value Attribute/Parameter is interpreted depends on the implementation<sup>5</sup>, and this document gives no constraints on the specification.

本項は、フィールドの特性（2.5.3.2 項で述べた固定長か可変長か、並びに、2.5.3.3 項で述べたビット長）とデータ型について定める。なお、データ型は、種別が Numerical Value または Binary Sequence である Attribute/Parameter に対してのみ定める。

Binary Sequence Attribute/Parameter の Data Type は、対応するフィールドが可変長か固定長かを指定するものである Binary Sequence Data Type である。Binary Sequence データ型と対応するフィールドの特性については、3.1.2.2 項参照。

Numerical Value Attribute/Parameter のデータ型は、計算機（宇宙機搭載と地上の双方）で工学値（2.5.5 項）を扱う方法を指定するものである Numerical Value データ型 である。Numerical Value データ型と対応するフィールドの特性については、3.1.2.3 項参照。

[注] Numerical Value Attribute/Parameter のデータ型の解釈方法は実装に依存し<sup>6</sup>、本書はその仕様を制約しない。

---

<sup>5</sup> In an on-board software, it is used as the type of the variable that holds the Raw Data Value of an Attribute, and it is used as the type of variable that holds the Engineering Value of an Attribute in a ground software.

<sup>6</sup> 搭載ソフトウェアにおいて、Attribute の生データ値を保持する変数の型として使用されたり、地上ソフトウェアにおいて Attribute の工学値を保持する変数の型に使用されたりしている。

If the type of the Attribute/Parameter is neither a Numerical Value nor Binary Sequence (*i.e.*, if it is an Enumerative or State), no Data Type is specified. The corresponding field (*i.e.*, Enumerative or State Attribute/Parameter field) shall have a fixed length between 1 and 32 bits and its Encoding Method shall be unsigned.

Attribute/Parameter の種別が Numerical Value でも Binary Sequence でもない場合（すなわち、Enumerative または State の場合）、データ型は定めない。この場合、対応するフィールド（つまり、Enumerative または State Attribute/Parameter フィールド）は 1 ビットから 32 ビットの固定長であること。また、エンコーディング方式は、unsigned であること。

### 3.1.2.2. Binary Sequence Data Type and corresponding fields

#### Binary Sequence データ型と対応するフィールド

If the type of the Attribute/Parameter is a Binary Sequence<sup>7</sup>, the Encoding Method shall not be specified for the corresponding field (*i.e.*, Binary Sequence Attribute/Parameter field) and the Data Type of the field shall be a Binary Sequence Data Type (either “variableLengthBinary” or “hexBinary”, depending on whether the field is in variable length or fixed length, respectively). Of the two Data Types, variableLengthBinary<sup>8</sup> shall be used only for the last field of an SMCP Message.

Attribute/Parameter の種別が Binary Sequence<sup>9</sup>である場合、対応するフィールド(つまり、Binary Sequence Attribute/Parameter フィールド)にはエンコーディング方式を指定しないこと。また、そのフィールドのデータ型は、Binary Sequence データ型(そのフィールドが可変長か固定長かに応じ、それぞれ、“variableLengthBinary” か “hexBinary” の何れか)であること。二つのデータ型のうち、variableLengthBinary<sup>10</sup> は、SMCP Message の最後のフィールドにのみ用いること。

---

<sup>7</sup> SIB2XML [R2] supports only a multiple of eight for the bit length of a Binary Sequence Attribute field.

<sup>8</sup> SIB2XML [R2] supports variableLengthBinary only for the ACTION Telecommand.

<sup>9</sup> SIB2XML [R2] は、Binary Sequence Attribute フィールドのビット長として、8の倍数のみをサポートしている。

<sup>10</sup> SIB2XML [R2] は、variableLengthBinary を ACTION Telecommand に対してのみサポートしている。

### 3.1.2.3. Numerical Value Data Type and corresponding fields

#### Numerical Value データ型と対応するフィールド

If the type of the **Attribute/Parameter** is a **Numerical Value**, the **Encoding Method** (Section 3.1.3) **shall** be specified for the corresponding field (*i.e.*, **Numerical Value Attribute/Parameter** field), the field **shall** be in fixed length, and the **Data Type** for the **Engineering Value** of the field **shall** be a **Numerical Value Data Type**, which is one of “byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, and “double” as defined in Informative Reference [R1].

Of the **Numerical Value Data Types**, **byte**, **unsignedByte**, **short**, **unsignedShort**, **int**, **unsignedInt**, **long**, and **unsignedLong** are **Integer Data Types**.

[Note] Table 3-1 shows the permitted range of the values for fields of the **Integer Data Types**. Of the **Numerical Value Data Types**, **float** and **double** are **Floating-Point Data Types**, which correspond to floating point numbers in single and double precisions, respectively.

The bit length of the **Numerical Value Attribute/Parameter** field **shall** be in the range from 1 to 64.

**Attribute/Parameter** の種別が **Numerical Value** である場合、対応するフィールド(つまり、**Numerical Value Attribute/Parameter** フィールド)には**エンコーディング方式**(3.1.3 項)を指定すること。また、そのフィールドは固定長であること。そのフィールドの**工学値**に対する**データ型**は**Numerical Value データ型**(参考文書 [R1] のデータ型に由来する“byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, または “double”の何れか)であること。

**Numerical Value データ型**のうち、**byte**, **unsignedByte**, **short**, **unsignedShort**, **int**, **unsignedInt**, **long**, 及び **unsignedLong**は**整数データ型**である。

[注] Table 3-1 に、**整数データ型**のフィールドの値の範囲を示す。

**Numerical Value データ型**のうち、**float** 及び **double**は**浮動小数点データ型**である。それぞれ、单精度及び倍精度浮動小数点数に対応する。

**Numerical Value Attribute/Parameter** フィールドのビット長は、1から64の範囲にあること。

Note that, if an **Engineering Unit Conversion** is used for decoding the value of a field of an **SMCP Message**, the **Data Type** **should** be **double**. [Rationale] All the known existing implementations can interpret any **Engineering Value** as a double-precision floating-point number although the identical **Engineering Value** might not be obtained in different implementations because how **Data Types** are handled is not specified in this document.

[Temporary Note] Here we would like to define the concept of a time array to deal with various telemetry-message-editing schemes. A time array is defined as  $[A(T-(n-1)d), A(T-(n-2)d), \dots, A(T-d), A(T-d), A(T)]$ , where  $A(t)$  is the value of **Attribute**  $A$  at time  $t$  and  $T$  is the **SMCP Message** reference time specified in Section 4.3 of [A3].

We would also like to allow “arrays” (as in the programming sense) of time arrays]

なお、**SMCP Message** のフィールドの値のデコードに**工学値変換**を用いる場合、**データ型**は**double**であるべきである。[根拠] 本書は**データ型**を如何に扱うかを定めておらず、実装が異なると同一の**工学値**が得られない事がある。他方で、既知の全ての実装は、**工学値**を倍精度浮動小数点数として解釈できるため。

[一時注] ここでは、様々なテレメトリメッセージ編集方式に対処するために時間配列の概念を定義したいと思います。時間配列は、 $A(t)$ を時刻  $t$ における **Attribute**  $A$  の値とした時に  $[A(T-(n-1)d), A(T-(n-2)d), \dots, A(T-d), A(T)]$  として定義されます。ここで、 $T$  は [A3] 4.3 項で規定されている **SMCP Message** 基準時刻です。また、(プログラミング言語における) 配列の時間配列も許したいと思います]

**Table 3-1 Value Ranges of the Integer Data Types // 整数データ型の値域**

Data Type	sign & bit length	Minimum Value // 最小値	Maximum Value // 最大値
<b>byte</b>	singed 8bit	-128	127
<b>unsignedByte</b>	unsinged 8bit	0	255
<b>short</b>	singed 16bit	-32768	32767
<b>unsignedShort</b>	unsinged 16bit	0	65535
<b>int</b>	singed 32bit	-2147483648	2147483647
<b>unsignedInt</b>	unsinged 32bit	0	4294967295
<b>long</b>	singed 64bit	-9223372036854775808	9223372036854775807
<b>unsignedLong</b>	unsinged 64bit	0	18446744073709551615

### 3.1.3. Encoding Method // エンコーディング方式

An Encoding Method (Section 2.5.4.2) **shall** be one of “`unsigned`”, “`signed`”, and “`ieee754`”.

If no Engineering Unit Conversion is specified for decoding the value of a field of an SMCP Message, the combination of an Encoding Method and Data Type **should** be one of those listed in Table 3-2. [Rationale] With these, the information would not be lost in Numerical Encoding or Numerical Decoding.

エンコーディング方式 (2.5.4.2 項) は、 “`unsigned`”, “`signed`”, または “`ieee754`” の何れかである**こと**。

SMCP Message のフィールドの値のデコードに 工学値変換 を定めない場合、エンコーディング方式とデータ型の組み合わせは Table 3-2 に示すものである**べき**である。[根拠] この組合せであれば、数値エンコードや数値デコードにおいて情報の欠損が起きない。

Table 3-2: Combination of Encoding Methods and Data Types without Engineering Unit Conversion // 工学値変換を用いない場合のエンコーディング方式とデータ型の組み合わせ

Encoding Method エンコーディング方式	Data Type データ型
absent 存在しない	<code>hexBinary</code> or <code>variableLengthBinary</code>
<code>unsigned</code>	<code>unsignedByte</code> , <code>unsignedShort</code> , <code>unsignedInt</code> , or <code>unsignedLong</code>
<code>signed</code>	<code>byte</code> , <code>short</code> , <code>int</code> , or <code>long</code>
<code>ieee754</code>	<code>float</code> or <code>double</code>

### 3.1.4. Condition Expressions // 条件式

In this section, an **Attribute** or **Pseudo Attribute** is simply referred to as an **Attribute**. A **Condition Expression** **shall** be either one of the **Comparison Terms** given in Table 3-3 or an arbitrary number of **Comparison Terms** connected with AND, OR, XOR, and/or NOT, where the operators are allowed to be nested up to four levels. In the following, each of A, B, C, ... represents a **Reference Name** of a referenced **Attribute**. Also, the constant in a **Comparison Term** **shall** be a value that the referenced **Attribute** is permitted to take.

本項では、**Attribute** または **Pseudo Attribute** の事を単に **Attribute** と称する。

**条件式**は、Table 3-3 に示す **Comparison Terms** の一つか、または、任意個の **Comparison Terms** を AND, OR, XOR, 及びまたは NOT で接続したもの（ここで、演算子の入れ子は四階層まで許容される）である**こと**。以下で、A, B, C, ... は、何れも、参照 **Attribute** の**参照名**を表すとする。また、**Comparison Term** に含まれる定数は、参照 **Attribute** が取り得る値である**こと**。

Table 3-3: Type of **Comparison Term**

**Comparison Term** の種類

<b>Comparison Term</b>	Explanations / 説明	Note / 備考
A == constant A == 定数	<b>Attribute</b> value is equal to the specified constant <b>Attribute</b> の値が指定した定数と等しい	
A != constant A != 定数	<b>Attribute</b> value is not equal to the specified constant <b>Attribute</b> の値が指定した定数に等しくない	
A > constant A > 定数	<b>Attribute</b> value is larger than the specified constant <b>Attribute</b> の値が指定した定数より大きい	Only applicable for <b>Numerical Value Attributes</b> and <b>Pseudo Attributes</b>
A >= constant A >= 定数	<b>Attribute</b> value is equal to or is larger than the specified constant // <b>Attribute</b> の値が指定した定数以上である	
A < constant A < 定数	<b>Attribute</b> value is smaller than the specified constant <b>Attribute</b> の値が指定した定数より小さい	<b>Numerical Value Attributes</b> 及び <b>Pseudo Attributes</b> にのみ適用可能
A <= constant A <= 定数	<b>Attribute</b> value is equal to or is smaller than the specified constant // <b>Attribute</b> の値が指定した定数以下である	

[Example]

((A == constant) AND (B > constant)) OR ((A == 定数) AND (B > 定数)) OR ((C ==  
(C == constant) AND (D > constant)) 定数) AND (D > 定数))

[例]

## 3.2. Functional Object Definition // Functional Object 定義

### 3.2.1. General // 一般

The functions that a spacecraft and its onboard instruments have are generally complex. Hence, it is a common practice to define groups of functions each of which is closely related. An entity that has a group of functions of this kind is called a [Functional Object](#) in the FMS (see Section 3.2 in [A2]).

An entire spacecraft is a [Functional Object](#). Any group of functions of arbitrary granularity such as subsystems, onboard instruments, and onboard software functions are regarded as [Functional Objects](#) depending upon the design of the spacecraft.

The [Functional Object Definition](#) is a bundle of definitions of [Functional Objects](#).

A [Functional Object Definition](#) **shall** define zero or more [Functional Objects](#).

一般に宇宙機やその搭載機器が持つ機能は複雑である。そこで、通例として密接に関連した機能のグループを定義する。このような一群の機能をもつ構成要素を、FMS では [Functional Object](#) と呼ぶ ([A2] 3.2 項参照)。

宇宙機全体は、[Functional Object](#) である。また、宇宙機の設計に応じ、そのサブシステム、搭載機器、搭載ソフトウェアの持つ機能など任意の粒度の機能の塊が [Functional Object](#) とみなされる。

[Functional Object 定義](#)は、単数または複数の [Functional Objects](#) の定義をまとめたものである。

[Functional Object 定義](#)は、ゼロ個以上の [Functional Objects](#) を定義すること。

A Functional Object contains zero or more Functional Objects in it. When a Functional Object contains Functional Objects in it, the former is called the “parent Functional Object” of the latter and the latter are called “child Functional Objects” of the former (see Section 3.2.2 in [A2]).

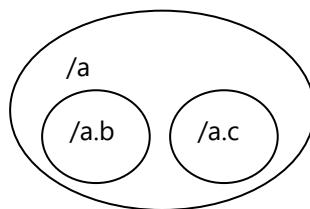
The top-level Functional Object of a spacecraft is a Functional Object that represents the entire spacecraft.

The parent-child relationships of Functional Objects shall be defined with their Absolute Names (see Figure 3-1).

Functional Object は、ゼロ個以上の Functional Objects を含む。ある Functional Object に Functional Objects が含まれる場合、前者は後者の「親 Functional Object」と呼ばれ、後者は前者の「子 Functional Objects」と呼ばれる ([A2] 3.2.2 項参照)。

ある宇宙機で最上位の Functional Object は、その宇宙機全体を表す Functional Object である。

Functional Objects の親子関係は、各々の絶対名にて定義すること (Figure 3-1 参照)。



**Figure 3-1: Parent-Child Relationship of Functional Objects**  
**Functional Objects の親子関係**

[Note] The names of the Functional Objects in Figure 3-1 are /a, /a.b, and /a.c where b and c are child Functional Objects whose parent Functional Object is a. For the exact definition of the parent-child relationships of Functional Objects, see Normative Reference [A2].

[注] Figure 3-1 の Functional Objects の絶対名は、/a, /a.b, 及び /a.c である。b 及び c は、a を親 Functional Object とする子 Functional Objects である。Functional Objects の親子関係の厳密な定義は、引用文書 [A2] 参照。

### 3.2.2. Functional Object

#### 3.2.2.1. General // 一般

For a **Functional Object**, the following items **shall** be defined:

[**Function Definition**]

- 1) **Functional Object** name  
::= **Absolute Name** of the **Functional Object**
- 2) **Effective-Condition** (optional)  
::= Condition for the **Functional Object** to function (see Section 3.2.3 in [A2]; it is either “**always**” or a **Reference Name** of a **Named Condition Expression**)

This item **shall** be specified for each of the **Functional Objects** aboard a spacecraft, except for the **Functional Object** representing the entire spacecraft, for which this item **shall** not be specified.

If the **Functional Object** always functions, **always shall** be specified for this item. Otherwise, a **Reference Name** of a **Named Condition Expression** (see Section 3.6.2.1) **shall** be specified for this item.

[Note] **Telecommand Issue SW** performs **Prechecks** (Section 2.6.3) according to this information, Section 3.2.5.4 **Function Definition** 2) **Effective-Condition**, and Section 3.3.2.3 **Function Definition** 2) **Begin State**.

- 3) **Attribute** (0~n)  
See Section 3.2.3.3
- 4) **Operation** (0~n)  
See Section 3.2.5.4
- 5) **Alert class** (0~n)
- 3) **Attribute** (0~n)  
3.2.3.3 項参照
- 4) **Operation** (0~n)  
3.2.5.4 項参照
- 5) **Alert class** (0~n)

各 **Functional Object** に、以下の項目を定義すること。

[**機能定義**]

- 1) **Functional Object** 名  
::= **Functional Object** の絶対名
- 2) **有効条件** (optional)  
::= **Functional Object** が機能する条件 ([A2] 3.2.3 項参照; **名前付条件式の参照名** または “**always**”)

本項目は、各 **Functional Object** に指定すること。ただし、宇宙機の全体を表す **Functional Object** には、本項目を指定しないこと。

**Functional Object** が常に機能する場合、**always** を本項目に指定すること。さもなければ、**名前付条件式の参照名** (3.6.2.1 項参照) を本項目に指定すること。

[注] **テレコマンド発行 SW** は、この情報、3.2.5.4 項**機能定義** 2)**有効条件**、及び 3.3.2.3 項**機能定義** 2)**Begin State** 名に従って、**Prechecks** (2.6.3 項) を実施する。

See Section 3.2.6.4

3.2.6.4 項参照

## [Information Definition]

- 1) **UFORID** (1~n) (Section 3.2.2.2)  
 $\text{::=}$  the identifier of the combination of the **Upper FOID** of the **Functional Object** (*i.e.*, the source of **Response VALUE Telemetries** (Section 3.2.3.2) and **NOTIFICATION Telemetries** (Section 3.2.6.2) and the destination of **ACTION Telecommands** (Section 3.2.5.2), **SET Telecommands**, and **GET Telecommands** (see Section 3.2.3.2) all of which are specified in the definition of this **Functional Object** in the **Functional Object Definition**) and the delivery **Route** of these **Telemetries** and **Telecommands**  
The number of the **UFORIDs** to which no **Route** name is specified **shall** be zero or one.

2) **Lower FOID**

$\text{::=}$  **Lower FOID** of the **Functional Object** (see Section 3.3.6 in [A3])

3) **ACK Response or Not**

$\text{::=}$  whether the **Functional Object** sends an **ACK Telemetry** or not in response to an **SMCP Telecommand** that requests an **ACK Telemetry** (either “**response**” or “**not**”)

4) **Pseudo Attribute** (0~n)

See Section 3.2.4.2

## [情報定義]

- 1) **UFORID** (1~n) (3.2.2.2 項)  
 $\text{::=}$  **Functional Object** (つまり、**Functional Object 定義**において本 **Functional Object** の定義が規定する **Response VALUE Telemetries** (3.2.3.2 項) 及び **NOTIFICATION Telemetries** (3.2.6.2 項) 全ての生成元、かつ、**ACTION Telecommands** (3.2.5.2 項), **SET Telecommands**, 及び **GET Telecommands** (3.2.3.2 項参照) 全ての配布先) の **Upper FOID**、並びに、これらの **Telemetries** 及び **Telecommands** の配送 **Route** の組み合わせの識別子

**Route** 名を指定しない **UFORIDs** の数は0か1であること。

2) **Lower FOID**

$\text{::=}$  **Functional Object** の **Lower FOID** ([A3] 3.3.6 項参照)

3) **ACK 応答有無**

$\text{::=}$  **Functional Object** が、**ACK Telemetry** 要求する **SMCP Telecommand** に対する応答として、**ACK Telemetry** を送出するか否か (“**response**” または “**not**” の何れか)

4) **Pseudo Attribute** (0~n)

3.2.4.2 項参照

### 3.2.2.2. UFORID

For a **UFORID**, the following items **shall** be defined:

[Information Definition]

1) **Value**

$\text{ ::= value of the UFORID } (\text{UFORID} = \text{Upper FOID} + \text{Route ID} = \text{Lower APID};$   
 $\text{ see Section 3.3.5 in [A3])}$

For a **Functional Object**, the value of the **Upper FOID** embedded in any of **UFORIDs** **should** be identical according to [A3].

The value of the **Route ID** embedded in a **UFORID** **should** be zero if no **Route** name is specified for the **UFORID**; [Rationale] for clarification.

2) **Route name (optional)**

$\text{ ::= Local Name of the Route}$

If the **UFORID** indicates that the delivery **Route** is not specified, this item **shall** not be specified. Otherwise, this item **shall** be specified.

各 **UFORID** に、以下の項目を定義すること。

[情報定義]

1) **値**

$\text{ ::= UFORID の値 } (\text{UFORID} = \text{Upper FOID} + \text{Route ID} = \text{Lower APID};$   
 $\text{ [A3] 3.3.5 項参照})$

[A3] に従い、ある **Functional Object** に対して、何れの **UFORID** に埋め込まれる **Upper FOID** の値も、同一であるべきである。

ある **UFORID** に **Route** 名を指定しない場合、その **UFORID** に埋め込まれる **Route ID** の値はゼロであるべきである；[根拠] 明確にするため。

2) **Route 名 (optional)**

$\text{ ::= Route の局所名}$

**UFORID** が配送 **Route** を明示しない事を示す場合、本項目は指定しないこと。さもなければ、本項目を指定すること。

## [Example]

In the case where the delivery **Route** is not explicitly specified (*i.e.*, if there is only one delivery **Route** or the spacecraft has the function to automatically select the delivery **Route**):

One **UFORID**:

**Route name:** None, **Route ID:** None

In the case where the delivery **Route** is explicitly specified:

Two **UFORIDs**:

**Route name:** A, **Route ID:** 0  
**Route name:** B, **Route ID:** 1

In the case where the delivery **Route** might or might not be explicitly specified:

Three **UFORIDs**:

**Route name:** None, **Route ID:** 0  
**Route name:** A, **Route ID:** 1  
**Route name:** B, **Route ID:** 2

[Note] If both of **UFORIDs** that indicate that the delivery **Route** is explicitly specified and a **UFORID** that indicates that the delivery **Route** is not explicitly specified are defined for a **Functional Object**, the spacecraft must have the ability to select automatically the delivery **Route**, referring to the **Route ID**.

## [例]

配送 **Route** を明示しない場合（つまり、配送 **Route** が一つしか存在しない場合、または、宇宙機が配送 **Route** を自動で選択する機能を持つ場合）：

**UFORID** が一つ：

**Route 名:** なし, **Route ID:** なし

配送 **Route** を明示する場合：

**UFORID** が二つ：

**Route 名:** A, **Route ID:** 0  
**Route 名:** B, **Route ID:** 1

配送 **Route** を明示する事、明示しない事の双方がある場合：

**UFORID** が三つ：

**Route 名:** なし, **Route ID:** 0  
**Route 名:** A, **Route ID:** 1  
**Route 名:** B, **Route ID:** 2

[注] ある **Functional Object** に対して、配送 **Route** を明示する事を示す **UFORIDs** と配送 **Route** を明示しない事を示す **UFORID** の双方が定義される場合、宇宙機は **Route ID** を参照し、配送 **Route** を自動で選択する機能を持つ必要がある。

### 3.2.3. Attribute Definition // Attribute 定義

#### 3.2.3.1. General //一般

An [Attribute](#) of a [Functional Object](#) is a parameter representing a status of the [Functional Object](#) (see Section 3.3 in [A2]).

An [Attribute](#) is one of [Binary Sequence Attribute](#), [Numerical Value Attribute](#), [Enumerative Attribute](#), and [State Attribute](#) (Sections 3.3.5, 3.3.7, 3.3.6, and 3.6.3, respectively, in [A2]).

Zero or more [Attributes](#) (and [Attribute](#) fields of related [Telecommands](#) and [Telemetries](#)) are defined for each of the [Functional Object](#) specified according to Section 3.2.2.1.

[Functional Object](#) の [Attribute](#) は、[Functional Object](#) の状態を表すパラメータである ([A2] 3.3 項参照)。

[Attribute](#) は、[Binary Sequence Attribute](#), [Numerical Value Attribute](#), [Enumerative Attribute](#), または [State Attribute](#) の何れかである (それぞれ、[A2] 3.3.5 項, 3.3.7 項, 3.3.6 項, 及び 3.6.3 項)。

3.2.2.1 項に従って定めた [Functional Objects](#) の各々に、[Attributes](#) (及び関連する [Telecommands](#) と [Telemetries](#) の [Attribute](#) フィールド) をゼロ個以上定義する。

#### 3.2.3.2. VALUE Telemetry, SET Telecommand, and GET Telecommand

The definition of an [Attribute](#), in some cases, includes the definitions of a format of [GET Telecommands](#) (Section 4.4 in [A3]) and a format of [Response VALUE Telemetries](#) (Section 5.3.1 in [A3]), and/or a format of [SET Telecommands](#) (Section 4.5 in [A3]) [Section 3.2.3.3 [Information Definition](#) 4) and 5)]. Note that each of the [Response VALUE Telemetries](#) and [SET Telecommands](#) of these contains only one sub-field for an [Attribute](#) in the [Attribute Values](#) field.

[Attribute](#) の定義は、場合により、[GET Telecommands](#) ([A3] 4.4 項) のフォーマットと [Response VALUE Telemetries](#) ([A3] 5.3.1 項) のフォーマット、及び / または、[SET Telecommands](#) ([A3] 4.5 項) のフォーマットの定義を含む [3.2.3.3 項 [情報定義](#) 4)及び 5)]. なお、これらのうち、[Response VALUE Telemetries](#) 及び [SET Telecommands](#) は、各々、[Attribute Values](#) フィールドに、[Attribute](#) に対するサブフィールドを一つのみ含む。

### 3.2.3.3. Attribute (field)

For an **Attribute** (field), the following items **shall** be defined:

[Function Definition]

- 1) **Attribute name**  
 $\text{ ::= Local Name of the Attribute}$
- 2) **Attribute Type**  
 $\text{ ::= the type of the Attribute (one of Binary Sequence, Numerical Value, Enumerative, and State)}$
- 3) **Data Type** (optional)  
 See Section 3.1.2.1 (one of “byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, “double”, “hexBinary”, and “variableLengthBinary”)
- 4) **Initial Value** (optional)  
 $\text{ ::= Initial Value of the Attribute (see Section 3.3.4 in [A2])}$
- 各 **Attribute** (フィールド) に、以下の項目を定義すること。  
**[機能定義]**
  - 1) **Attribute 名**  
 $\text{ ::= Attribute の局所名}$
  - 2) **Attribute 種別**  
 $\text{ ::= Attribute の種別 (Binary Sequence, Numerical Value, Enumerative, または State の何れか)}$
  - 4) **データ型** (optional)  
 3.1.2.1 項参照 (“byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, “double”, “hexBinary”, または “variableLengthBinary” の何れか)
  - 5) **Initial Value** (optional)  
 $\text{ ::= Attribute の Initial Value ([A2] 3.3.4 項参照)}$

[Information Definition]

1) Effective-Condition

$::=$  Condition requiring that the value of the **Attribute** field be valid (see Section 3.3.2 in [A2]; “**always**” or a **Reference Name** of a **Named Condition Expression**)

If the value of the **Attribute** field is always valid, **always shall** be specified for this item. Otherwise, a **Reference Name** of a **Named Condition Expression** (see Section 3.6.2.1) **shall** be specified for this item.

2) isVariableLengthOrNot

$::=$  whether the **Attribute** field is in variable length or not (see Section 2.5.3.2; either “**variable\_length**” (meaning variable length) or “**not**” (fixed length))<sup>11</sup>

The **Attribute** field is in fixed length unless it is located at the end of the **SMCP Message** (see Section 3.1.2.2).

[Note] If the **Data Type** is **variableLengthBinary**, the value of this item is **variable\_length**. Otherwise, **not**.

3) Bit-Length (1~2)

$::=$  the bit length (if in fixed length) or maximum bit length (if in variable length) of the **Attribute** field (see Section 2.5.3.3) defined in a **VALUE Telemetry** (including **Response VALUE Telemetry**) (Sections 3.2.3.2 and 3.7.2), **NOTIFICATION Telemetry** (Section 3.2.6.2), **ACTION Telecommand** (Section 3.2.5.2), or **SET Telecommand** (Sections 3.2.3.2 and 3.7.2)

[情報定義]

1) 有効条件

$::=$  **Attribute** フィールドの値が有効である条件 ([A2] 3.3.2 項参照; **名前付条件式の参照名** または “**always**”)

**Attribute** フィールドの値が常に有効な場合、**always** を本項目に指定すること。さもなければ、**名前付条件式の参照名** (3.6.2.1 項参照) を本項目に指定すること。

2) 可変長か否か

$::=$  **Attribute** フィールドが可変長か否か (2.5.3.2 項参照; “**variable\_length**” (可変長) または “**not**” (固定長) の何れか)<sup>12</sup>

**Attribute** フィールドは、同 **SMCP Message** の末尾にある場合を除き、固定長である (3.1.2.2 項参照)。

[注] **データ型**が**variableLengthBinary**の場合、本項目の値は**variable\_length**である。さもなければ、**not**である。

3) Bit-Length (1~2)

$::=$  **VALUE Telemetry** (**Response VALUE Telemetry** を含む) (3.2.3.2 項及び 3.7.2 項), **NOTIFICATION Telemetry** (3.2.6.2 項), **ACTION Telecommand** (3.2.5.2 項), または **SET Telecommand** (3.2.3.2 項及び 3.7.2 項) に定義する **Attribute** フィールドのビット長 (固定長の場合) または最大ビット長 (可変長の場合) (2.5.3.3 項参照)

<sup>11</sup> SIB2XML [R2] does not support specifying **variable\_length** for this item.

<sup>12</sup> SIB2XML [R2] は、本項目に **variable\_length** を指定する事をサポートしていない。

If only one **Bit-Length** is specified, the bit lengths of the **Attribute** fields contained in the **Telemetries** and **Telecommands** are identical. By specifying two **Bit-Lengths**, it is possible to allow different bit lengths depending on how the **Telemetries** and **Telecommands** are defined (see Section 2.5.3.3). In this case, the first one (called “**tight**”) is for a **VALUE Telemetry** (including a **Response VALUE Telemetry**) and **SET Telecommand** either for which multiple fields are specified to the **Attribute Values** field by means of the method described in Section 3.7.5, and the second one (called “**loose**”) is for (1) an **ACTION Telecommand** (Section 3.2.5.2), (2) a **NOTIFICATION Telemetry** (Section 3.2.6.2), and (3) a **Response VALUE Telemetry** and **SET Telecommand** either for which only one field is specified to the **Attribute Values** field by means of the method described in Section 3.2.3.2.

3) **Encoding Method** (optional)

See Section 3.1.3 (one of “**unsigned**”, “**signed**”, and “**ieee754**”)

4) **isGettableOrNot**

::= whether a format of **GET Telecommands** and a format of **Response VALUE Telemetries** are specified for the **Attribute** or not (either “**gettable**” or “**not**”)

5) **isSettableOrNot**

::= whether a format of **SET Telecommands** are specified for the **Attribute** or not (either “**settable**” or “**not**”)

**Bit-Length** を一つのみ指定した場合、**Telemetries** 及び **Telecommands** に含める該当 **Attribute** フィールドのビット長は同一である。**Bit-Length** を二つ指定する事で、**Telemetries** 及び **Telecommands** の定義法に応じてビット長を変える事ができる(2.5.3.3 項参照)。この場合、一つ目 (“**tight**”と呼ぶ) は 3.7.5 項に記した方法で **Attribute Values** フィールドに対して複数のフィールドを定める **VALUE Telemetry** (**Response VALUE Telemetry** を含む) 及び **SET Telecommand** 向けのものであり、二つ目 (“**loose**” と呼ぶ) は、(1) **ACTION Telecommand** (3.2.5.2 項) 、 (2) **NOTIFICATION Telemetry** (3.2.6.2 項) 、及び、(3) 3.2.3.2 項に記した方法で **Attribute Values** フィールドに対して一つのみのフィールドを定める **Response VALUE Telemetry** 及び **SET Telecommand** 向けのものである。

3) **エンコーディング方式** (optional)

3.1.3 項参照 (“**unsigned**”, “**signed**”, または “**ieee754**”の何れか)

4) **GET 可能か否か**

::= **Attribute** に対して、**GET Telecommands** のフォーマットと **Response VALUE Telemetries** のフォーマットを規定するか否か (“**gettable**” または “**not**” の何れか)

5) **SET 可能か否か**

::= **Attribute** に対して、**SET Telecommands** のフォーマットを規定するか否か (“**settable**” または “**not**” の何れか)

- |  |   |
|--|---|
| <p>6) <b>Attribute ID</b> (optional)<br/> <math>\coloneqq</math> the identifier of the <b>Attribute</b> in the <b>Response VALUE Telemetry</b>, <b>SET Telecommand</b>, or <b>GET Telecommand</b> (see Section 5.3.3 in [A3])<br/> If <b>isGettableOrNot</b> is <b>gettable</b> or <b>isSettableOrNot</b> is <b>settable</b> (<i>i.e.</i>, if <b>GET Telecommands</b> and/or <b>SET Telecommands</b> are specified), this item <b>shall</b> be specified.<br/> <br/> [Note] The <b>Attribute IDs</b> of <b>Attributes</b> and <b>Attribute Sequences</b> share the common ID space (see Section 3.7.5.1 2)).</p> | <p>6) <b>Attribute ID</b> (optional)<br/> <math>\coloneqq</math> Response <b>VALUE Telemetry</b>, <b>SET Telecommand</b>, または <b>GET Telecommand</b> の <b>Attribute</b> の識別子 ([A3] 5.3.3 項参照)<br/> <br/> GET 可能か否かが <b>gettable</b> または <b>SET 可能か否か</b>が <b>settable</b> である場合 (つまり、<b>GET Telecommands</b> と <b>SET Telecommands</b> の一方または双方を規定する場合)、本項目を指定すること。<br/> [注] <b>Attributes</b> と <b>Attribute Sequences</b> の <b>Attribute IDs</b> は、共通の ID 空間を共有する (3.7.5.1 項 2)参照)。</p> |
| <p>7) <b>Upper APID</b><br/> <math>\coloneqq</math> the entities that are allowed to monitor the <b>Response VALUE Telemetry</b> (see Section 5.2.3 in [A4]; either “<b>any</b>” (meaning “monitored by <b>Controllers</b> and <b>Monitors</b>”) or “<b>monitor_only</b>” (“monitored by <b>Monitors</b> only”))</p>   | <p>7) <b>Upper APID</b><br/> <math>\coloneqq</math> Response <b>VALUE Telemetry</b> の監視を許容する構成要素 ([A4] 5.2.3 項参照; “<b>any</b>”: <b>Controllers</b> と <b>Monitors</b> によって監視される、または“<b>monitor_only</b>”: <b>Monitors</b>のみによって監視される、の何れか)</p>   |

### 3.2.3.4. Binary Sequence Attribute (field)

For a **Binary Sequence Attribute** (field), the definition items for the **Attribute** (field) are specified, which satisfy the following conditions<sup>13</sup> (see Section 3.1.2.2).

The **Data Type** of a **Binary Sequence Attribute** is either “**variableLengthBinary**” or “**hexBinary**” and the corresponding **Binary Sequence Attribute** field is in variable length or fixed length, respectively.

No **Encoding Method** is specified for a **Binary Sequence Attribute** field.

Only one **Bit-Length** **shall** be specified for a **Binary Sequence Attribute** field.

**Binary Sequence Attribute** (フィールド)には、**Attribute** (フィールド)に対する定義項目を定める。その際に、以下の制約<sup>14</sup>を満たす(3.1.2.2 項参照)。

**Binary Sequence Attribute** のデータ型は、“**variableLengthBinary**” または “**hexBinary**” の何れかであり、対応する **Binary Sequence Attribute** フィールドは、それぞれ、可変長または固定長である。

**Binary Sequence Attribute** フィールドにエンコーディング方式は指定しない。

**Binary Sequence Attribute** フィールドの **Bit-Length** は、一つのみ指定すること、

---

<sup>13</sup> SIB2XML [R2] supports only a multiple of eight for the bit length of a **Binary Sequence Attribute** field.

<sup>14</sup> SIB2XML [R2] は、**Binary Sequence Attribute** フィールドのビット長として、8の倍数のみをサポートしている。

### 3.2.3.5. Numerical Value Attribute (field)

For a **Numerical Value Attribute** (field), the definition items for the **Attribute** (field) are specified, which satisfy the following conditions (see Section 3.1.2.3).

The **Data Type** of a **Numerical Value Attribute** is either one of **byte**, **unsignedByte**, **short**, **unsignedShort**, **int**, **unsignedInt**, **long**, **unsignedLong**, **float**, and **double** if specified or might be left unspecified<sup>15</sup>.

A **Numerical Value Attribute** field is in fixed length.

The bit length of a **Numerical Value Attribute** field is in the range from 1 to 64.

For a **Numerical Value Attribute** (field), the following items **shall** also be defined:

#### [Function Definition]

- 1) **Valid Range**: **Upper Limit** (optional), **Lower Limit** (optional)

See Section 3.3.7.2 in [A2]

If this item is specified, **Telecommand Issue SW** validates the values of the parameters of **ACTION Telecommands** and **SET Telecommands** (see Section 2.6.2) according to this information.

Not specifying an **Upper Limit** or **Lower Limit** means that no upper or lower limits are present, respectively.

**Numerical Value Attribute** (フィールド)には、**Attribute** (フィールド)に対する定義項目を定める。その際に、以下の制約を満たす(3.1.2.3 項参照)。

**Numerical Value Attribute** のデータ型は、**byte**, **unsignedByte**, **short**, **unsignedShort**, **int**, **unsignedInt**, **long**, **unsignedLong**, **float**, または **double** の何れかを指定するか、何も指定しない<sup>16</sup>か、の何れかである。

**Numerical Value Attribute** フィールドは、固定長である。

**Numerical Value Attribute** フィールドのビット長は、1 から 64 の範囲にある。

各 **Numerical Value Attribute** (フィールド)に、以下の項目も定義すること。

#### [機能定義]

- 1) **Valid Range**: **上限値** (optional), **下限値** (optional)

[A2] 3.3.7.2 項参照

本項目が指定された場合、**テレコマンド発行 SW** は、本情報に従って、**ACTION Telecommands** と **SET Telecommands** のパラメータの値を検証(2.6.2 項参照)する。

**上限値**を指定しない事は上限が無い事を意味する。**下限値**を指定しない事は下限が無い事を意味する。

---

<sup>15</sup> This item is mandatory in the current implementation as of March 2023.

<sup>16</sup> 2023 年 3 月現在の実装では本項目は必須である。

- 2) Attribute Limit name (optional)  
 ::= Reference Name of the Attribute Limit  
 (see Section 3.5.2.1) for the Limit Checks  
 If this item is specified, the Data Conversion Module performs Limit Checks (Section 2.6.5) according to the Attribute Limit Definition.
- 2) Attribute Limit 名 (optional)  
 ::= Limit Checks する Attribute Limit の参照名 (3.5.2.1 項参照)  
 本項目が指定された場合、データ変換モジュールは、Attribute Limit 定義に従って、Limit Checks (2.6.5 項) を実施する。

## [Information Definition]

- 1) Decode Engineering Unit Conversion name (optional)  
 ::= Reference Name of the Decode Engineering Unit Conversion (Section 3.8.2.1)  
 If the Data Conversion Module performs a Decode Engineering Unit Conversion (Section 2.5.5.4) for the value of the Numerical Value Attribute field of the VALUE Telemetry or NOTIFICATION Telemetry, this item shall be specified. Otherwise, this item shall not be specified.
- 2) Encode Engineering Unit Conversion name (optional)  
 ::= Reference Name of the Encode Engineering Unit Conversion (see Section 3.8.2.1)  
 If Telecommand Issue SW performs an Encode Engineering Unit Conversion (Section 2.5.5.3) for the value of the Numerical Value Attribute field of the ACTION Telecommand or SET Telecommand, this item shall be specified. Otherwise, this item shall not be specified.

## [情報定義]

- 1) Decode 工学値変換名 (optional)  
 ::= Decode 工学値変換 (3.8.2.1 項) の参照名  
 データ変換モジュールが、VALUE Telemetry または NOTIFICATION Telemetry の同 Numerical Value Attribute フィールドの値に Decode 工学値変換 (2.5.5.4 項) を実施する場合、本項目を指定すること。さもなければ、本項目は指定しないこと。
- 2) Encode 工学値変換名 (optional)  
 ::= Encode 工学値変換の参照名 (3.8.2.1 項参照)  
 テレコマンド発行 SW が、ACTION Telecommand または SET Telecommand の同 Numerical Value Attribute フィールドの値に Encode 工学値変換 (2.5.5.3 項) を実施する場合、本項目を指定すること。さもなければ、本項目は指定しないこと。



### 3.2.3.6. **Enumerative Attribute** (field)

For an **Enumerative Attribute** (field), the definition items for the **Attribute** (field) are specified, which satisfy the following conditions (see Section 3.1.2.1).

No **Data Type** is specified for an **Enumerative Attribute**.

An **Enumerative Attribute** field is in fixed length.

The bit length of an **Enumerative Attribute** field is in the range from 1 to 32.

The **Encoding Method** of an **Enumerative Attribute** field is **unsigned**.

For an **Enumerative Attribute** (field), the following item **shall** also be defined:

[**Function Definition**]

1) **Valid Value Set name**

::= **Reference Name** of the **Valid Value Set** (Section 3.9.2.1) that specifies the values that the **Enumerative Attribute** takes

**Enumerative Attribute** (フィールド) には、**Attribute** (フィールド) に対する定義項目を定める。その際に、以下の制約を満たす (3.1.2.1 項参照)。

**Enumerative Attribute** には、**データ型**を指定しない。

**Enumerative Attribute** フィールドは、固定長である。

**Enumerative Attribute** フィールドのビット長は、1 から 32 の整数の範囲にある。

**Enumerative Attribute** フィールドの**エンコーディング方式**は、**unsigned** である。

各 **Enumerative Attribute** (フィールド) に、以下の項目も定義すること。

[**機能定義**]

1) **Valid Value Set 名**

::= **Enumerative Attribute** が取る値を指定する **Valid Value Set** (3.9.2.1 項) の**参照名**

### 3.2.3.7. State Attribute (field)

For a **State Attribute** (field), the definition items for the **Attribute** (field) are specified, which satisfy the following conditions (see Section 3.1.2.1).

No **Data Type** is specified for a **State Attribute**.

A **State Attribute** field is in fixed length.

The bit length of a **State Attribute** field is in the range from 1 to 32.

The **Encoding Method** of a **State Attribute** field is **unsigned**.

The **isSettable** of a **State Attribute** **should** be **false**; [Rationale] in order to define the behavior of the **State Attribute** solely with the **State Machine definition**.

For a **State Attribute** (field), the following item **shall** also be defined:

[Function Definition]

1) **State Machine name**

$::=$  **Reference Name** of the **State Machine** corresponding to the **State Attribute** (Section 3.6.3 in [A2]) (see Section 3.3.2.1)

**State Attribute** (フィールド) には、**Attribute** (フィールド) に対する定義項目を定める。その際に、以下の制約を満たす (3.1.2.1 項参照)。

**State Attribute** には、**データ型**を指定しない。  
**State Attribute** フィールドは、固定長である。  
**State Attribute** フィールドのビット長は、1 から 32 の範囲にある。

**State Attribute** フィールドの**エンコーディング方式**は、**unsigned** である。

**State Attribute** の **SET 可能**は、**false** であるべきである; [根拠] **State Attribute** の振る舞いを **State Machine 定義**のみで定義するため。

各 **State Attribute** (フィールド) に、以下の項目も定義すること。

[機能定義]

1) **State Machine 名**

$::=$  **State Attribute** に対応する **State Machine** ([A2] 3.6.3 項) の**参照名** (3.3.2.1 項参照)

### 3.2.4. Pseudo Attribute Definition // Pseudo Attribute 定義

#### 3.2.4.1. General // 一般

The value of a **Pseudo Attribute** is an **Engineering Value**, which is calculated from other **Engineering Values**. A **Pseudo Attribute** is handled in the same way as an **Attribute** in application programs that handle **Engineering Values** (see Section 2.5.8).

Zero or more **Pseudo Attributes** are defined for each of the **Functional Objects** specified according to Section 3.2.2.1.

[Note] The **Data Conversion Module** calculates the values of **Pseudo Attributes** according to the information specified in this section.

**Pseudo Attribute** の値は、他の（時には複数の）**工学値**から計算される**工学値**であり、**Pseudo Attribute** は、**工学値**を扱うアプリケーションプログラムにおける **Attribute** と同様に扱われる（2.5.8 項参照）。

3.2.2.1 項に従って定めた **Functional Objects** の各々に、**Pseudo Attributes** をゼロ個以上定義する。

[注] **データ変換モジュール**は、本項で指定する情報に従って、**Pseudo Attributes** の値を計算する。

#### 3.2.4.2. Pseudo Attribute

For a **Pseudo Attribute**, the following items **shall** be defined:

##### [Information Definition]

- 1) **Pseudo Attribute** name  
::= Local Name of the **Pseudo Attribute**
- 2) **Effective-Condition**  
::= Condition for updating the value of the **Pseudo Attribute** at the **Calculation Timing** (“**always**” or a **Reference Name** of a **Named Condition Expression**)

If the update of the value of the **Pseudo Attribute** is always valid, **always shall** be specified for this item. Otherwise, a **Reference Name** of a **Named Condition Expression** (see Section 3.6.2.1) **shall** be specified for this item.

各 **Pseudo Attribute** に、以下の項目を定義すること。

##### [情報定義]

- 1) **Pseudo Attribute** 名  
::= **Pseudo Attribute** の局所名
- 2) **有効条件**  
::= 計算タイミングに **Pseudo Attribute** の値の更新を実施する条件（名前付条件式の参照名または“**always**”）

**Pseudo Attribute** の値の更新が常に有効な場合、**always** を本項目に指定すること。さもなくば、名前付条件式の参照名（3.6.2.1 項参照）を本項目に指定すること。

- 3) **Engineering Unit Conversion name**  
 ::= Reference Name of the **Engineering Unit Conversion** (see Section 3.8.2.1) that yields the value of the **Pseudo Attribute**
- 4) **Calculation Timing** (1~n)  
 ::= Reference Name of the **Attribute Sequence** (see Section 3.7.5.1) for which the value of the **Pseudo Attribute** is calculated  
 The **Data Conversion Module** calculates the value of the **Pseudo Attribute** at the timing of receiving a **Telemetry Message** that carries the **Attribute Sequence**.
- 5) **Attribute Limit name (optional)**  
 ::= Reference Name of the **Attribute Limit** (see Section 3.5.2.1) for the **Limit Checks**  
 If this item is specified, The **Data Conversion Module** performs **Limit Checks** (Section 2.6.5) according to the **Attribute Limit Definition**.
- 3) **工学値変換名**  
 ::= Pseudo Attribute の値を得る**工学値変換の参照名** (3.8.2.1 項参照)
- 4) **計算タイミング** (1~n)  
 ::= その Pseudo Attribute の値を計算する **Attribute Sequence の参照名** (3.7.5.1 項参照)  
 データ変換モジュールは、同 **Attribute Sequence** を運ぶ **Telemetry Message** の受信のタイミングで、Pseudo Attribute の値を計算する。
- 5) **Attribute Limit 名 (optional)**  
 ::= **Limit Checks** する **Attribute Limit の参照名** (3.5.2.1 項参照)  
 本項目が指定された場合、データ変換モジュールは、**Attribute Limit 定義**に従って、**Limit Checks** (2.6.5 項) を実施する。

### 3.2.5. Operation Definition // Operation 定義

#### 3.2.5.1. General //一般

An [Operation](#) of a [Functional Object](#) is an action performed by the [Functional Object](#) and is invoked from the outside (see Section 3.4 in [A2]).

Zero or more [Operations](#) are defined for each of the [Functional Objects](#) specified according to Section 3.2.2.1.

[Functional Object](#) の [Operation](#) は、[Functional Object](#) が行う動作であり、外部より呼び出す ([A2] 3.4 項参照)。

3.2.2.1 項に従って定めた [Functional Objects](#) の各々に、[Operations](#) をゼロ個以上定義する。

#### 3.2.5.2. ACTION Telecommand

The definition of an [Operation](#) (see Section 3.2.5.4) includes the definitions of a format of [ACTION Telecommands](#) (Section 4.3 in [A3]) that invoke the [Operation](#). Here, the definition of the sequence of the sub-fields of the [Parameters and Attribute Values](#) field of an [ACTION Telecommand](#) uses [ACTION Telecommand Field Elements](#) ([ACT Field Elements](#); Section 3.2.5.3) and might use also, if required, [Blocks](#) (Section 3.2.5.5).

A [Block](#) is a grouping of a sequence of fields. In the definition of the [Operation](#), an [Internal Block](#) (Section 3.2.5.3 [Function Definition/Information Definition](#) 1)) might be specified as a sub-field. The [Internal Block](#) is defined inside the [Operation](#) definition.

ある [Operation](#) の定義 (3.2.5.4 項参照) は、その [Operation](#) を呼び出す [ACTION Telecommands](#) ([A3] 4.3 項) のフォーマットの定義を含む。ここで、[ACTION Telecommand](#) の [Parameters and Attribute Values](#) フィールドのサブフィールドの並びの定義においては [ACTION Telecommand Field Elements](#) ([ACT Field Elements](#); 3.2.5.3 項)、また、必要に応じ、[Blocks](#) (3.2.5.5 項) を用いる。

[Block](#) はフィールドの並びをグループ化したものである。[Operation](#) の定義では、サブフィールドとして [内部 Block](#) (3.2.5.3 項 [機能定義・情報定義](#) 1)) を指定する事ができる。[内部 Blocks](#) はその [Operation](#) 定義の内部で定義される。

### 3.2.5.3. ACTION Telecommand Field Element (ACT Field Element)

An ACTION Telecommand Field Element (ACT Field Element) defines a sub-field of the Parameters and Attribute Values field of an ACTION Telecommand. An ACT Field Element is one of the following items:

#### [Function Definition]

- 1) ParameterValueSetting Attribute name (attributeRef)

$\coloneqq$  Reference Name of the Value Setting Attribute (\*\*1)

\*\*1: Attribute whose value is set to the value of a sub-field of the Parameters and Attribute Values field of an ACTION Telecommand (see Section 3.4.4.1 in [A2])

[Note] Telecommand Issue SW performs Success Verifications (Section 2.6.4.5) according to this information.

- 2) Parameters (field) (parameter)

$\coloneqq$  sub-field that does not specify the values of Attributes

#### [Information Definition]

- 1) Constant field (constant)

$\coloneqq$  A sub-field that does has a fixed value (Using this as a part of the key that identifies the Operation is deprecated because it does not comply with the SMCP).

A project **may** determine that a Constant field be acceptable as the first parameter, whose field locates immediately after the Operation ID field. In this case, Constant field can be regarded as an extension of the Operation ID.

ACTION Telecommand Field Element (ACT Field Element) は、ACTION Telecommand の Parameters and Attribute Values フィールドのサブフィールドを定義する。ACT Field Element は、以下の項目の何れかである。

#### [機能定義]

- 1) パラメータ値設定 Attribute 名 (attributeRef)

$\coloneqq$  Value Setting Attribute (\*\*1) の参照名

\*\*1: ACTION Telecommand の Parameters and Attribute Values フィールドのサブフィールドの値に値を設定する Attribute ([A2] 3.4.4.1 項参照)

[注] テレコマンド発行 SW は、この情報に従って、Success Verifications (2.6.4.5 項)を実施する。

- 2) Parameter (フィールド) (parameter)

$\coloneqq$  サブフィールドで Attributes の値を指定しないもの

#### [情報定義]

- 1) 定数 フィールド (constant)

$\coloneqq$  サブフィールドで固定の値を持つもの (Operation を識別するキーの一部として用いる事は SMCP を逸脱するため非推奨)。

Project は、先頭のパラメータとして、定数 フィールドを、許容すると決めて**良い**。先頭のパラメータのフィールドは、Operation ID フィールドの直後に位置し、Operation ID を拡張するものとみなす事ができる。

- |  |   |
|--|---|
| 2) <b>Reserved</b> field ( <b>reserve</b> )<br>::= a sub-field that has no functional meaning  | 2) <b>予約</b> フィールド ( <b>reserve</b> )<br>::= サブフィールドで機能的な意味を持たないもの  |
| <b>[Function Definition/Information Definition]</b>  |   |
| 1) <b><u>Internal Block</u></b> ( <b>block</b> )<br>See Section 3.2.5.5<br><br>An <b>Internal Block</b> is allowed to be defined in the definition of the <b>Operation</b> . | 1) <b>内部 Block</b> ( <b>block</b> )<br>3.2.5.5 項参照<br><br><b>Operation</b> の定義の中で <b>内部 Block</b> を定義する事ができる。 |

### 3.2.5.4. Operation and ACTION Telecommand

For an **Operation** and an **ACTION Telecommand** that invokes the **Operation**, the following items **shall** be defined:

#### [Function Definition]

##### 1) **Operation** name

**::= Local Name** of the **Operation**

The **Local Name** of an **Operation** **should** not begin with **SET\_** or **GET\_**; [Rationale] in order to identify easily from the **Telecommand Name** (Section 5.4) whether it is an **ACTION Telecommand**, **SET Telecommand**, or **GET Telecommand**.

##### 2) **Effective-Condition**

**::= Condition** for the **Operation** to function (see Section 3.4.5 in [A2]; “**always**” or a **Reference Name** of a **Named Condition Expression**)

If the **Operation** can always be invoked, **always shall** be specified for this item. Otherwise, a **Reference Name** of a **Named Condition Expression** (see Section 3.6.2.1) **shall** be specified for this item.

[Note] **Telecommand Issue SW** performs **Prechecks** (Section 2.6.3) according to this information, Section 3.2.2.1 **Function Definition** 2) **Effective-Condition**, and Section 3.3.2.3 **Function Definition** 2) **Begin State**.

各 **Operation** 及びその **Operation** を呼び出す各 **ACTION Telecommand** に、以下の項目を定義すること。

#### [機能定義]

##### 1) **Operation** 名

**::= Operation の局所名**

**Operation の局所名**は **SET\_**または **GET\_**から始まるべきではない；[根拠] **Telecommand 名** (5.4 項) から **ACTION Telecommand**, **SET Telecommand**, または **GET Telecommand** の何れなのかが容易に分かるようにするため。

##### 2) **有効条件**

**::= Operation が機能する条件** ([A2] 3.4.5 項参照； **名前付条件式の参照名** または “**always**”)

**Operation** を常に呼び出す事ができる場合、**always**を本項目に指定すること。さもなければ、**名前付条件式の参照名** (3.6.2.1 項参照) を本項目に指定すること。

[注] **テレコマンド発行 SW** は、この情報、3.2.2.1 項**機能定義** 2)**有効条件**、及び 3.3.2.3 項**機能定義** 2)**Begin State** 名に従って、**Prechecks** (2.6.3 項) を実施する。

- 3) **Attribute Change Rule name** (optional)  
 $::=$  Reference Name of an **Attribute Change Rule** (see Section 3.4.2.1)
- If this item is specified, **Telecommand Issue SW** performs **Success Verifications** (Section 2.6.4.5), according to the **Attribute Change Rules**, for **ACTION Telecommands** that invoke the **Operation**.
- [Note] A **State** transitions as a result of the execution of an **Operation** is defined as a **State Transition class** in the **State Machine definition** (see Section 3.3.2.3) without using the **Attribute Change Rules**.
- 4) **Criticality Level**  
 $::=$  **Criticality Level** of the **Operation** (see Section 3.4.6 in [A2]; one of “**prohibited**”, “**warning**”, and “**normal**”)
- [Note1] **Telecommand Issue SW** informs the **Criticality Level** to its human operator if the **Criticality Level** is not **normal** (see Section 2.6.2).
- [Note2] **Telecommand Issue SW** does not issue an **Action Telecommand** whose **Criticality Level** is **prohibited**.
- [Information Definition]
- 1) **Operation ID**  
 $::=$  the identifier of the **Operation** that is invoked by the **ACTION Telecommand** (see Section 4.2.2 in [A3])
- 3) **Attribute Change Rule** 名 (optional)  
 $::=$  **Attribute Change Rule** の 参 照 名  
(3.4.2.1 項参照)
- 本項目が指定された場合、**テレコマンド発行 SW** は、**Operation** を呼び出す **ACTION Telecommands** に 対 し て 、 **Attribute Change Rules** に 従 つ て 、 **Success Verifications** (2.6.4.5 項) を 実 施 す る。
- [注] **Operation** の 実 行 の 結 果 と し て の **State** の 遷 移 は 、 **Attribute Change Rules** は 用 い ず 、 **State Machine 定 義** お い て **State Transition class** と し て 定 義 す る (3.3.2.3 項 参 照)。
- 4) **Criticality Level**  
 $::=$  **Operation** の **Criticality Level** ([A2] 3.4.6 項 参 照; “**prohibited**”, “**warning**”, ま た は “**normal**” の 何 れ か)
- [注 1] **テレコマンド発行 SW** は、**Criticality Level** が **normal** で な い 場 合 、 本 情 報 に 従 つ て 、 そ の **Criticality Level** を 操 作 者 に 知 ら せ る (2.6.2 項 参 照)。
- [注 2] **テレコマンド発行 SW** は、**Criticality Level** が **prohibited** の **Action Telecommand** は 発 行 し な い。
- [情報定義]
- 1) **Operation ID**  
 $::=$  **ACTION Telecommand** が 呼 び 出 す **Operation** の 識 別 子 ([A3] 4.2.2 項 参 照)

2) **Upper APID**

::= the entities that are allowed to issue the **ACTION telecommand** (see Section 5.2.3 in [A4]; one of “**onboard\_only**” (meaning “issued only from inside the spacecraft”), “**ground\_only**” (“issued only from the ground”), and “**both**” (both are allowed))

[Note] **Telecommand Issue SW** checks, according to this information, whether the entity is allowed to issue a **Telecommand** (see Section 2.6.2).

## [Function Definition/Information Definition]

1) **ACT Field Element** (0~n)

See Section 3.2.5.3

The definition order of the **ACT Field Elements** **shall** be the same as the order of the sub-fields of a **Parameters and Attribute Values** field of the **ACTION Telecommand**, including the nested structure that contains **Blocks**.

The bit position (Section 2.5.1) of a **constant** field (Section 3.2.5.10) **shall** be a multiple of four; [Rationale] in order to ensure the readability of **ACTION Telecommands** in the hexadecimal notation.

2) **Upper APID**

::= **ACTION Telecommand** の発行を許容する構成要素 ([A4] 5.2.3 項参照; “**onboard\_only**”: 宇宙機内部からのみ発行される、“**ground\_only**”: 地上からのみ発行される、または “**both**”: それらの双方が許容される、の何れか)

[注] **テレコマンド発行 SW** は、本情報に従って、構成要素が **Telecommands** を発行する事が許容されるかチェックする (2.6.2 項参照)。

## [機能定義・情報定義]

1) **ACT Field Element** (0~n)

## 3.2.5.3 項参照

**ACT Field Elements** の定義順は、**Blocks** を含む入れ子構造も含めて、**ACTION Telecommand** の **Parameters and Attribute Values** フィールドのサブフィールドの並び順と同一であること。

**定数** フィールド (3.2.5.10 項) のビット位置 (2.5.1 項) は、4 の倍数であること; [根拠] 16 進数表記の **ACTION Telecommands** の可読性を確保するため。

### 3.2.5.5. Internal Block // 内部 Block

For an Internal Block, the following items shall be defined:

[Function Definition]

- 1) Block name (optional)

::= Local Name of the Block

[Function Definition/Information Definition]

- 1) ACT Field Element (0~n)

See Section 3.2.5.3

ACT Field Elements except for Internal Blocks are allowed to be nested.

各内部 Block に、以下の項目を定義すること。

[機能定義]

- 1) Block 名 (optional)

::= Block の局所名

[機能定義・情報定義]

- 1) ACT Field Element (0~n)

3.2.5.3 項参照

内部 Blocks 以外の ACT Field Elements の入れ子が可能である。

### 3.2.5.6. Parameter (field)

A **Parameter** of the **Operation** (see Section 3.4.4 in [A2]) **shall** be one of **Binary Sequence Parameter**, **Numerical Value Parameter**, and **Enumerative Parameter**.

For each **Parameter** of the **Operation** and each **Parameter** field of the **ACTION Telecommand**, the following items **shall** be defined:

#### [Function Definition]

- 1) **Parameter name**  
 ::= Local Name of the Parameter
- 2) **Parameter Type**  
 ::= the type of the Parameter (one of **Binary Sequence**, **Numerical Value**, and **Enumerative**)
- 3) **Data Type** (optional)  
 See Section 3.1.2.1 (one of “byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, “double”, “hexBinary”, and “variableLengthBinary”)

**Operation** の **Parameter** ([A2] 3.4.4 項参照) は、**Binary Sequence Parameter**, **Numerical Value Parameter** または **Enumerative Parameter** の何れかである**こと**。

**Operation** の各 **Parameter** 及び **ACTION Telecommand** の各 **Parameter** フィールドに、以下の項目を定義する**こと**。

#### [機能定義]

- 1) **Parameter 名**  
 ::= Parameter の局所名
- 2) **Parameter 種別**  
 ::= Parameter の種別 (**Binary Sequence**, **Numerical Value**, または **Enumerative** の何れか)
- 3) **データ型** (optional)  
 3.1.2.1 項参照 (“byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, “double”, “hexBinary”, または “variableLengthBinary” の何れか)

## [Information Definition]

1) isVariableLengthOrNot

$\coloneqq$  whether the **Parameter** field is in variable length or not (see Section 2.5.3.2; either “variable\_length” (meaning variable length) or “not” (fixed length))

The **Parameter** field is in fixed length unless it is located at the end of a **Telecommand Message** (see Section 3.1.2.2).

[Note] If the **Data Type** is **variableLengthBinary**, the value of this item is **variable\_length**. Otherwise, **not**.

2) Bit-Length

$\coloneqq$  the bit length (if in fixed length) or maximum bit length (if in variable length) of the **Parameter** field (see Section 2.5.3.3)

3) Encoding Method (optional)

See Section 3.1.3 (one of “**unsigned**”, “**signed**”, and “**ieee754**”)

## [情報定義]

1) 可変長か否か

$\coloneqq$  **Parameter** フィールドが可変長か否か (2.5.3.2 項参照; “variable\_length” (可変長) または “not” (固定長) の何れか)

**Parameter** フィールドは、**Telecommand Message** の末尾にある場合を除き、固定長である (3.1.2.2 項参照)。

[注] **データ型**が **variableLengthBinary** の場合、本項目の値は **variable\_length** である。さもなければ、**not** である。

2) Bit-Length

$\coloneqq$  **Parameter** フィールドのビット長 (固定長の場合) または最大ビット長 (可変長の場合) (2.5.3.3 項参照)

3) エンコーディング方式 (optional)

3.1.3 項参照 (“**unsigned**”, “**signed**”, または “**ieee754**”の何れか)

### 3.2.5.7. Binary Sequence Parameter (field)

For a **Binary Sequence Parameter** (field), the definition items for the **Parameter** (field) are specified, which satisfy the following conditions<sup>17</sup> (see Section 3.1.2.2).

The **Data Type** of a **Binary Sequence Parameter** is either “**variableLengthBinary**” or “**hexBinary**” and the corresponding **Binary Sequence Parameter** field is in variable length or fixed length, respectively.

No **Encoding Method** is specified for a **Binary Sequence Parameter** field.

**Binary Sequence Parameter** (フィールド)には、**Parameter** (フィールド)に対する定義項目を定める。その際に、以下の制約<sup>18</sup>を満たす(3.1.2.2 項参照)。

**Binary Sequence Parameter** のデータ型は、“**variableLengthBinary**” または “**hexBinary**” の何れかであり、対応する **Binary Sequence Parameter** フィールドは、それぞれ、可変長または固定長である。

**Binary Sequence Parameter** フィールドにエンコーディング方式は指定しない。

---

<sup>17</sup> SIB2XML [R2] supports only a multiple of eight for the bit length of a **Binary Sequence Attribute** field.

<sup>18</sup> SIB2XML [R2] は、**Binary Sequence Attribute** フィールドのビット長として、8の倍数のみをサポートしている。

### 3.2.5.8. Numerical Value Parameter (field)

For a **Numerical Value Parameter** (field), the definition items for the **Parameter** (field) are specified, which satisfy the following conditions (see Section 3.1.2.3).

The **Data Type** of a **Numerical Value Parameter** is either one of `byte`, `unsignedByte`, `short`, `unsignedShort`, `int`, `unsignedInt`, `long`, `unsignedLong`, `float`, and `double` if specified or might be left unspecified<sup>19</sup>.

A **Numerical Value Parameter** field is in fixed length.

The bit length of a **Numerical Value Parameter** field is in the range from 1 to 64.

**Numerical Value Parameter** (フィールド)には、**Parameter** (フィールド)に対する定義項目を定める。その際に、以下の制約を満たす(3.1.2.3 項参照)。

**Numerical Value Parameter** のデータ型は、`byte`, `unsignedByte`, `short`, `unsignedShort`, `int`, `unsignedInt`, `long`, `unsignedLong`, `float`, または `double` の何れかを指定するか、何も指定しない<sup>20</sup>か、の何れかである。

**Numerical Value Parameter** フィールドは、固定長である。

**Numerical Value Parameter** フィールドのビット長は、1 から 64 の範囲にある。

---

<sup>19</sup> This item is mandatory in the current implementation as of March 2023.

<sup>20</sup> 2023 年 3 月現在の実装では本項目は必須である。

For a Numerical Value Parameter (field), the following items **shall** also be defined:

[Function Definition]

- 1) Valid Range: Upper Limit (optional), Lower Limit (optional))

$\coloneqq$  valid range of the value of the parameter

If this item is specified, Telecommand Issue SW checks the values of the parameters of ACTION Telecommands (see Section 2.6.2) according to this information.

[Information Definition]

- 1) Encode Engineering Unit Conversion name (optional)

$\coloneqq$  Reference Name of the Encode Engineering Unit Conversion (Section 3.8.2.1)

If Telecommand Issue SW performs an Encode Engineering Unit Conversion (Section 2.5.5.3) for the value of the Numerical Value Parameter field of the ACTION Telecommand, this item **shall** be specified. Otherwise, this item **shall** not be specified.

各 Numerical Value Parameter (フィールド) に、以下の項目も定義すること。

[機能定義]

- 1) Valid Range: 上限値 (optional), 下限値 (optional)

$\coloneqq$  パラメータの値の有効範囲

本項目が指定された場合、テレコマンド発行 SW は、本情報に従って、ACTION Telecommands のパラメータの値をチェック (2.6.2 項参照) する。

[情報定義]

- 1) Encode 工学値変換名 (optional)

$\coloneqq$  Encode 工学値変換 (3.8.2.1 項) の参照名

テレコマンド発行 SW が、ACTION Telecommand の同 Numerical Value Parameter フィールドの値に Encode 工学値変換 (2.5.5.3 項) を実施する場合、本項目を指定すること。さもなければ、本項目は指定しないこと。

### 3.2.5.9. Enumerative Parameter (field)

For an **Enumerative Parameter** (field), the definition items for the **Parameter** (field) are specified, which satisfy the following conditions (see Section 3.1.2.1).

No **Data Type** is specified for an **Enumerative Parameter**.

An **Enumerative Parameter** field is in fixed length.

The bit length of an **Enumerative Parameter** field is in the range from 1 to 32.

The **Encoding Method** of an **Enumerative Parameter** field is **unsigned**.

For an **Enumerative Parameter** (field), the following item **shall** also be defined:

[**Function Definition**]

1) **Valid Value Set name**

::= **Refence Name** of the **Valid Value Set** (Section 3.9.2.1) that specifies the values that the **Enumerative Parameter** takes

**Enumerative Parameter** (フィールド) には、**Parameter** (フィールド) に対する定義項目を定める。その際に、以下の制約を満たす (3.1.2.1 項参照)。

**Enumerative Parameter** には、**データ型**を指定しない。

**Enumerative Parameter** フィールドは、固定長である。

**Enumerative Parameter** フィールドのビット長は、1 から 32 の範囲にある。

**Enumerative Parameter** フィールドの**エンコーディング方式**は、**unsigned** である。

各 **Enumerative Parameter** (フィールド) に、以下の項目も定義すること。

[**機能定義**]

1) **Valid Value Set 名**

::= **Enumerative Parameter** が取る値を指定する **Valid Value Set** (3.9.2.1 項) の**参照名**

### 3.2.5.10. Constant field // 定数フィールド

For a **constant** field, the following items **shall** be defined:

[Information Definition]

1) Bit-Length

The Bit-Length **shall** be a multiple of four.

2) Value

各定数フィールドに、以下の項目を定義すること。

[情報定義]

1) Bit-Length

Bit-Length は 4 の倍数であること。

2) 値

### 3.2.5.11. Reserved field // 予約フィールド

For a **reserved** field, the following item **shall** be defined:

[Information Definition]

1) Bit-Length

各予約フィールドに、以下の項目を定義すること。

[情報定義]

1) Bit-Length

### 3.2.6. Alert class Definition // Alert class 定義

#### 3.2.6.1. General //一般

An [event](#) of a [Functional Object](#) is an occurrence of a thing that has a particular significance. An [event](#) might be detected by the [Functional Object](#) itself. An [event](#) that is important to outside entities might be reported to them, and a classification of the report is called an [Alert class](#) (see Section 3.5.2 in [A2]).

Zero or more [Alert classes](#) are defined for each of the [Functional Object](#) specified according to Section 3.2.2.1.

[Functional Object](#) の [event](#) は、特定の意味を有する出来事の発生である。ある [event](#) は、[Functional Object](#) 自体が検出する事がある。また、外部の構成要素にとり重要な [event](#) を、それらに通知する事もある。この通知の分類を [Alert class](#) と呼ぶ ([A2] 3.5.2 項参照)。

3.2.2.1 項に従って定めた [Functional Object](#) の各々に、[Alert classes](#) をゼロ個以上定義する。

#### 3.2.6.2. NOTIFICATION Telemetry

The definition of an [Alert class](#) (see Section 3.2.6.4) includes the definitions of a format of [NOTIFICATION Telemetries](#) (Section 5.4 in [A3]) that report [Alerts](#) that belong to the [Alert class](#). Here, [NOTIFICATION Telemetry Field Elements](#) ([NTF Field Elements](#); Section 3.2.6.3) are used to define the sequence of the sub-fields of a [Parameters and Attribute Values](#) field of the [NOTIFICATION Telemetry](#).

ある [Alert class](#) の定義 (3.2.6.4 項参照) は、その [Alert class](#) に属する [Alerts](#) を通知する [NOTIFICATION Telemetries](#) ([A3] 5.4 項) のフォーマットの定義を含む。ここで、[NOTIFICATION Telemetry](#) の [Parameters and Attribute Values](#) フィールドのサブフィールドの並びを定義するのに [NOTIFICATION Telemetry Field Elements](#) ([NTF Field Elements](#); 3.2.6.3 項) を用いる。

### 3.2.6.3. NOTIFICATION Telemetry Field Elements (NTF Field Elements)

A NOTIFICATION Telemetry Field Element (NTF Field Element) defines a sub-field of the Parameters and Attribute Values field of a NOTIFICATION Telemetry. An NTF Field Element is one of the following items:

#### [Function Definition]

- 1) Value Notifying Attribute name (attributeRef)

$\coloneqq$  Reference Name of the Value Notifying Attribute (\*\*1)

\*\*1: Attribute whose value is notified of through a sub-field of the Parameters and Attribute Values field of a Notification Telemetry (see Section 3.5.2.3 in [A2])

- 2) Parameter (field) (parameter)

$\coloneqq$  a sub-field that does not have the values of Attributes

#### [Information Definition]

- 1) Reserved field (reserve)

$\coloneqq$  a sub-field that has no functional meaning

NOTIFICATION Telemetry Field Element (NTF Field Element) は、NOTIFICATION Telemetry の Parameters and Attribute Values フィールドのサブフィールドを定義する。NTF Field Element は、以下の項目の何れかである。

#### [機能定義]

- 1) Value Notifying Attribute name (attributeRef)  
 $\coloneqq$  Value Notifying Attribute (\*\*1) の参照名

\*\*1: Notification Telemetry の Parameters and Attribute Values フィールドのサブフィールドを通じて値を通知する Attribute ([A2] 3.5.2.3 項参照)

- 2) Parameter (フィールド) (parameter)

$\coloneqq$  サブフィールドのうち Attributes の値を持たないもの

#### [情報定義]

- 1) 予約フィールド (reserve)

$\coloneqq$  サブフィールドで機能的な意味を持たないもの

### 3.2.6.4. Alert class and NOTIFICATION Telemetry

For an **Alert class** and a **NOTIFICATION Telemetry** which reports an **Alert** that belongs to the **Alert class**, the following items **shall** be defined:

#### [Function Definition]

- 1) **Alert class name**  
::= Local Name of the **Alert class**
- 2) **Event class name**  
::= Reference Name of the **Event class** (Section 3.6.2.1) that contains the **Event** that triggers an **Alert** which belongs to the **Alert class**

#### [Information Definition]

- 1) **Alert ID**  
::= the identifier of the **Alert class** to which the **Alert**, reported by the **NOTIFICATION Telemetry**, belongs to (see Section 5.4.3 in [A3])
- 2) **Upper APID**  
::= the entities that are allowed to monitor the **NOTIFICATION Telemetry** (see Section 5.2.3 in [A4]; either “**any**” (meaning “monitored by **Controllers** and **Monitors**”) or “**monitor\_only**” (“monitored by **Monitors only**”))

#### [Function Definition/Information Definition]

- 1) **NTF Field Element (0~n)**

See Section 3.2.6.3

The order of the definitions of the **NTF Field Elements** **shall** be the same as the order of the sub-fields of the **Parameters and Attribute Values** field of the **NOTIFICATION Telemetry**.

各 **Alert class** 及びその **Alert class** に属する **Alert** を通知する各 **NOTIFICATION Telemetry** に、以下の項目を定義すること。

#### [機能定義]

- 1) **Alert class 名**  
::= **Alert class** の局所名
- 2) **Event class 名**  
::= **Alert class** に属する **Alert** を発生させる **Event** を含む **Event class** (3.6.2.1 項) の参照名

#### [情報定義]

- 1) **Alert ID**  
::= **NOTIFICATION Telemetry** が通知する **Alert** が属する **Alert class** の識別子 ([A3] 5.4.3 項参照)
- 2) **Upper APID**  
::= **NOTIFICATION Telemetry** の監視を許容する構成要素 ([A4] 5.2.3 項参照; “**any**”: **Controllers** と **Monitors** によって監視される、または“**monitor\_only**”: **Monitors**のみによって監視される、の何れか)

#### [機能定義・情報定義]

- 1) **NTF Field Element (0~n)**

#### 3.2.6.3 項参照

**NTF Field Elements** の定義順は、**NOTIFICATION Telemetry** の **Parameters and Attribute Values** フィールドのサブフィールドの並び順と同一であること。

### 3.2.6.5. Parameter (field)

A **Parameter** of the **Alert Class** (see Section 3.5.2.3 in [A2]) **shall** be one of **Binary Sequence Parameter**, **Numerical Value Parameter**, and **Enumerative Parameter**.

For each **Parameter** in the **Alert class** and each **Parameter** field of the **NOTIFICATION Telemetry**, the following items **shall** be defined:

#### [Function Definition]

- 1) **Parameter name**  
 ::= Local Name of the Parameter
- 2) **Parameter Type**  
 ::= the type of the Parameter (one of **Binary Sequence**, **Numerical Value**, and **Enumerative**)
- 3) **Data Type** (optional)  
 See Section 3.1.2.1 (one of “byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, “double”, “hexBinary”, and “variableLengthBinary”)

**Alert class** の **Parameter** ([A2] 3.5.2.3 項参照) は、**Binary Sequence Parameter**, **Numerical Value Parameter** または **Enumerative Parameter** の何れかである**こと**。

**Alert class** の各 **Parameter** 及び **NOTIFICATION Telemetry** の各 **Parameter** フィールドに、以下の項目を定義する**こと**。

#### [機能定義]

- 1) **Parameter 名**  
 ::= Parameter の局所名
- 2) **Parameter 種別**  
 ::= Parameter の種別 (**Binary Sequence**, **Numerical Value**, または **Enumerative** の何れか)
- 3) **データ型** (optional)  
 3.1.2.1 項参照 (“byte”, “unsignedByte”, “short”, “unsignedShort”, “int”, “unsignedInt”, “long”, “unsignedLong”, “float”, “double”, “hexBinary”, または “variableLengthBinary” の何れか)

## [Information Definition]

1) isVariableLengthOrNot

::= whether the **Attribute** field is in variable length or not (see Section 2.5.3.2; either “**variable\_length**” (meaning variable length) or “**not**” (fixed length))<sup>21</sup>

The **Parameter** field is in fixed length unless it is located at the end of a **Telemetry Message** (see Section 3.1.2.2).

[Note] If the **Data Type** is **variableLengthBinary**, the value of this item is **variable\_length**. Otherwise, **not**.

2) Bit-Length

::= the bit length (if in fixed length) or maximum bit length (if in variable length) of the **Parameter** field (see Section 2.5.3.3)

3) Encoding Method (optional)

See Section 3.1.3 (one of “**unsigned**”, “**signed**”, and “**ieee754**”)

## [情報定義]

1) 可変長か否か

::= **Parameter** フィールドが可変長か否か (2.5.3.2 項参照; “**variable\_length**” (可変長) または“**not**” (固定長) の何れか)<sup>22</sup>

**Parameter** フィールドは、**Telemetry Message** の末尾にある場合を除き、固定長である (3.1.2.2 項参照)。

[注] **データ型**が**variableLengthBinary**の場合、本項目の値は**variable\_length**である。さもなくば、**not**である。

2) Bit-Length

::= **Parameter** フィールドのビット長 (固定長の場合) または最大ビット長 (可変長の場合) (2.5.3.3 項参照)

3) エンコーディング方式 (optional)

3.1.3 項参照 (“**unsigned**”, “**signed**”, または “**ieee754**” の何れか)

<sup>21</sup> SIB2XML [R2] does not support specifying **variable\_length** for this item.

<sup>22</sup> SIB2XML [R2] は、本項目に **variable\_length** を指定する事をサポートしていない。

### 3.2.6.6. Binary Sequence Parameter (field)

For a **Binary Sequence Parameter** (field), the definition items for the **Parameter** (field) are specified, which satisfy the following conditions<sup>23</sup> (see Section 3.1.2.2).

The **Data Type** of a **Binary Sequence Parameter** is either “**variableLengthBinary**” or “**hexBinary**” and the corresponding **Binary Sequence Parameter** field is in variable length or fixed length, respectively.

No **Encoding Method** is specified for a **Binary Sequence Parameter** field.

**Binary Sequence Parameter** (フィールド)には、**Parameter** (フィールド)に対する定義項目を定める。その際に、以下の制約<sup>24</sup>を満たす(3.1.2.2 項参照)。

**Binary Sequence Parameter** のデータ型は、“**variableLengthBinary**” または “**hexBinary**” の何れかであり、対応する **Binary Sequence Parameter** フィールドは、それぞれ、可変長または固定長である。

**Binary Sequence Parameter** フィールドにエンコーディング方式は指定しない。

---

<sup>23</sup> SIB2XML [R2] does not support only a multiple of eight for the bit length of a **Binary Sequence Attribute** field.

<sup>24</sup> SIB2XML [R2] は、**Binary Sequence Attribute** フィールドのビット長として、8の倍数のみをサポートしている。

### 3.2.6.7. Numerical Value Parameter (field)

For a **Numerical Value Parameter** (field), the definition items for the **Parameter** (field) are specified, which satisfy the following conditions (see Section 3.1.2.3).

The **Data Type** of a **Numerical Value Parameter** is either one of **byte**, **unsignedByte**, **short**, **unsignedShort**, **int**, **unsignedInt**, **long**, **unsignedLong**, **float**, and **double** if specified or might be left unspecified<sup>25</sup>.

A **Numerical Value Parameter** field is in fixed length.

The bit length of a **Numerical Value Parameter** field is in the range from 1 to 64.

For a **Numerical Value Parameter** (field), the following item **shall** also be defined:

#### [Information Definition]

- 1) **Decode Engineering Unit Conversion name** (optional)

**::= Reference Name of the Decode Engineering Unit Conversion** (Section 3.8.2.1)

If the **Data Conversion Module** performs a **Decode Engineering Unit Conversion** (Section 2.5.5.4) for the value of the **Numerical Value Parameter** field of the **NOTIFICATION Telemetry**, this item **shall** be specified. Otherwise, this item **shall** not be specified.

**Numerical Value Parameter** (フィールド)には、**Parameter** (フィールド)に対する定義項目を定める。その際に、以下の制約を満たす(3.1.2.3 項参照)。

**Numerical Value Parameter** の**データ型**は、**byte**, **unsignedByte**, **short**, **unsignedShort**, **int**, **unsignedInt**, **long**, **unsignedLong**, **float**, または **double** の何れかを指定するか、何も指定しない<sup>26</sup>か、の何れかである。

**Numerical Value Parameter** フィールドは、固定長である。

**Numerical Value Parameter** フィールドのビット長は、1 から 64 の範囲にある。

各 **Numerical Value Parameter** (フィールド)に、以下の項目も定義すること。

#### [情報定義]

- 1) **Decode 工学値変換名** (optional)

**::= Decode 工学値変換** (3.8.2.1 項) の参照名

**データ変換モジュール**が、**NOTIFICATION Telemetry** の 同 **Numerical Value Parameter** フィールドの値に **Decode 工学値変換** (2.5.5.4 項) を実施する場合、本項目を指定すること。さもなければ、本項目は指定しないこと。

---

<sup>25</sup> This item is mandatory in the current implementation as of March 2023.

<sup>26</sup> 2023 年 3 月現在の実装では本項目は必須である。

### 3.2.6.8. Enumerative Parameter (field)

For an **Enumerative Parameter** (field), the definition items for the **Parameter** (field) are specified, which satisfy the following conditions (see Section 3.1.2.1).

No **Data Type** is specified for an **Enumerative Parameter**.

An **Enumerative Parameter** field is in fixed length.

The bit length of an **Enumerative Parameter** field is in the range from 1 to 32.

The **Encoding Method** of an **Enumerative Parameter** field is **unsigned**.

For an **Enumerative Parameter** (field), the following item **shall** also be defined:

[**Function Definition**]

1) **Valid Value Set name**

::= **Reference Name** of a **Valid Value Set** (Section 3.9.2.1) that specifies the values that the **Enumerative Parameter** takes

**Enumerative Parameter** (フィールド) には、**Parameter** (フィールド) に対する定義項目を定める。その際に、以下の制約を満たす (3.1.2.1 項参照)。

**Enumerative Parameter** には、**データ型**を指定しない。

**Enumerative Parameter** フィールドは、固定長である。

**Enumerative Parameter** フィールドのビット長は、1 から 32 の範囲にある。

**Enumerative Parameter** フィールドの**エンコーディング方式**は、**unsigned** である。

各 **Enumerative Parameter** (フィールド) に、以下の項目も定義すること。

[**機能定義**]

1) **Valid Value Set 名**

::= **Enumerative Parameter** が取る値を指定する **Valid Value Set** (3.9.2.1 項) の**参照名**

### 3.2.6.9. Reserved field // 予約フィールド

For a **reserved** field, the following item **shall** be defined:

[**Information Definition**]

1) **Bit-Length**

各 **予約** フィールドに、以下の項目を定義すること。

[**情報定義**]

1) **Bit-Length**

### 3.3. State Machine Definition // State Machine 定義

#### 3.3.1. General // 一般

A Behavior of a Functional Object describes rules about expected actions taken by the Functional Object, and is used to determine whether sequences of Operations are correct or not, whether Attributes are valid or not, etc. A Behavior of a Functional Object can be specified with a set of State Machines (Section 3.6 in [A2]).

The State Machine Definition is the definition to specify State Machines for Functional Objects.

A State Machine Definition shall define zero or more State Machines for a Functional Object.

Functional Object の Behavior は、Functional Object の期待される動作の規則を記すものであり、Operations の並びが正しいか否かや、Attributes の値が有効であるか否か等、を決めるのに用いる。Functional Object の Behavior は、State Machines ([A2] 3.6 項) の組により定める事ができる。

State Machine 定義は、Functional Objects に State Machines を定める定義である。

State Machine 定義は、各 Functional Object に State Machines をゼロ個以上定義すること。

### 3.3.2. State Machine

#### 3.3.2.1. General // 一般

For a **State Machine**, the following items **shall** be defined:

[Function Definition]

- 1) **Functional Object** name  
 ::= **Absolute Name** of the **Functional Object** that has the **State Machine**
- 2) **State Machine** name  
 ::= **Local Name** of the **State Machine**
- 3) **State** (1~n)  
 See Section 3.3.2.2
- 4) **State Transition class** (1~n)  
 See Section 3.3.2.3
- 5) **Effective-Condition**<sup>27</sup>  
 ::= Condition for the **State Machine** to function (“**always**” or a **Reference Name** of a **Named Condition Expression**)  
 If the **State Machine** always functions, **always shall** be specified for this item.  
 Otherwise, a **Reference Name** of a **Named Condition Expression** (see Section 3.6.2.1) **shall** be specified for this item.

各 **State Machine** に、以下の項目を定義すること。

[機能定義]

- 1) **Functional Object** 名  
 ::= **State Machine** を持つ **Functional Object** の絶対名
- 2) **State Machine** 名  
 ::= **State Machine** の局所名
- 3) **State** (1~n)  
 3.3.2.2 項参照
- 4) **State Transition class** (1~n)  
 3.3.2.3 項参照
- 5) **有効条件**<sup>28</sup>  
 ::= **State Machine** が機能する条件 (**名前付条件式の参照名** または “**always**”)

**State Machine** が常に機能する場合、**always** を本項目に指定すること。さもなければ、**名前付条件式の参照名** (3.6.2.1 項参照) を本項目に指定すること。

---

<sup>27</sup> SIB2XML [R2] supports only **always**.

<sup>28</sup> SIB2XML [R2] は、**always** のみをサポートしている。

### 3.3.2.2. State

For a **State** (Section 3.6.2 in [A2]), the following items **shall** be defined:

#### [Function Definition]

- 1) **State name**  
::= Local Name of the **State**

- 2) **Initial State** or not.

See Section 3.6.5 in [A2] (either “initial\_state” or “not”)

In the **State Machine**, the number of the **States** that are designated as an **Initial State** **shall** be one or zero.

- 3) **Criticality Level**

::= **Criticality Level** of the **State** (see Section 3.6.3 in [A2]; one of **Normal**, **Caution**, **Action**, and color names (“blue”, “magenta”, “cyan”, and “white”))

[Note] **Telemetry Monitor SW** displays the **State** name in the color according to the **Criticality Level** (see Section 2.6.6.3). Note that **Normal**, **Caution** and **Action** are mapped to colors “white”, “yellow” and “red”, respectively.

#### [Information Definition]

- 1) **Raw Data Value**

::= **Raw Data Value** to be encoded and set to telemetry fields

各 **State** ([A2] 3.6.2 項) に、以下の項目を定義すること。

#### [機能定義]

- 1) **State 名**  
::= **State の局所名**

- 2) **Initial State** か否か

[A2] 3.6.5 項参照 (“initial\_state” か “not” の何れか)

ある **State Machine** で、**Initial State** であると指定される **States** の数は 1 か 0 であること。

- 3) **Criticality Level**

::= **State の Criticality Level** ([A2] 3.6.3 項参照； **Criticality Level** は、**Normal**, **Caution**, **Action**, または色名 (“blue”, “magenta”, “cyan”, または “white”))

[注] **テlemetry 監視 SW** は、**State** 名を、**Criticality Level** に従う色で画面に表示する (2.6.6.3 項参照)。ここで、**Normal**, **Caution** 及び **Action** は、それぞれ、“white”, “yellow” 及び “red” の色にマッピングされる。

#### [情報定義]

- 1) **生データ値**

::= エンコードして **テlemetry** のフィールドに設定する際の **生データ値**

### 3.3.2.3. State Transition class

For a **State Transition class** (Section 3.6.6 in [A2]), the following items **shall** be defined:

[**Function Definition**]

- 1) **Trigger class** (1~n)

See Section 3.3.2.4

- 2) **Begin State name**

$::=$  **Local Name** of the source **State** in a transition (see Section 3.6.6.1 in [A2])

[Note] **Telecommand Issue SW** performs **Prechecks** (Section 2.6.3) according to this information, Section 3.2.2.1 **Function Definition** 2) **Effective-Condition**, and Section 3.2.5.4 **Function Definition** 2) **Effective-Condition**.

- 3) **End State name**

$::=$  **Local Name** of the destination **State** in a transition (see Section 3.6.6.1 in [A2])

[Note] **Telecommand Issue SW** performs **Success Verifications** (Section 2.6.4.5) according to this information.

各 **State Transition class** ([A2] 3.6.6 項) に、以下の項目を定義すること。

[**機能定義**]

- 1) **Trigger class** (1~n)

3.3.2.4 項参照

- 2) **Begin State 名**

$::=$  遷移元の **State** ([A2] 3.6.6.1 項参照) の **局所名**

[注] **テレコマンド発行 SW** は、この情報、3.2.2.1 項**機能定義** 2) **有効条件**、3.2.5.4 項**機能定義** 2) **有効条件**に従って、**Prechecks** (2.6.3 項) を実施する。

- 3) **End State 名**

$::=$  遷移先の **State** ([A2] 3.6.6.1 項参照) の **局所名**

[注] **テレコマンド発行 SW** は、この情報に従って、**Success Verifications** (2.6.4.5 項) を実施する。

- |   |   |
|---|---|
| <p>4) <b>Maximum Allowable Transition Time</b><br/>(optional)</p> <p>::= the maximum time allowed for a transition (see Section 3.6.6.3 in [A2])</p> <p>If this item is specified, <b>Telecommand Issue SW</b> performs <b>Success Verifications</b> (Section 2.6.4.5) according to this information.</p> <p>If the <b>Minimum Allowable Transition Time</b> is specified, this item <b>shall</b> be specified.</p> | <p>4) <b>Maximum Allowable Transition Time</b><br/>(optional)</p> <p>::= 遷移に許容される最大時間 ([A2] 3.6.6.3 項参照)</p> <p>本項目が指定された場合、<b>テレコマンド発行 SW</b> は、この情報に従って、<b>Success Verifications</b> (2.6.4.5 項) を実施する。</p> <p><b>Minimum Allowable Transition Time</b> を指定する場合、本項目を指定すること。</p> |
| <p>5) <b>Minimum Allowable Transition Time</b><br/>(optional)</p> <p>::= the minimum time allowed for a transition (see Section 3.6.6.3 in [A2])</p> <p>If this item is specified, <b>Telecommand Issue SW</b> performs <b>Success Verifications</b> (Section 2.6.4.5) according to this information.</p>   | <p>5) <b>Minimum Allowable Transition Time</b><br/>(optional)</p> <p>::= 遷移に許容される最小時間 ([A2] 3.6.6.3 項参照)</p> <p>本項目が指定された場合、<b>テレコマンド発行 SW</b> は、この情報に従って、<b>Success Verifications</b> (2.6.4.5 項) を実施する。</p>   |

### 3.3.2.4. Trigger class

For a **Trigger class** (Section 3.6.6.2 in [A2]), the following items **shall** be defined:

[Function Definition]

- 1) **Trigger class type**  
 $\coloneqq$  the type of a **Trigger class** (see Section 3.6.6.2 in [A2]; one of “**command**”, “**event**”, and “**spontaneous**”).
- 2) **Trigger class Name (optional)**  
 $\coloneqq$  the name of the **Trigger class**  
If the **Trigger class** type is **command** or **event**, a **Reference Name** of the trigger **Operation** (Section 3.2.5.4) or a **Reference Name** of the trigger **Event class** (Section 3.6.2.1), respectively, **shall** be specified for this item. Otherwise, this item **shall** not be specified.

各 **Trigger class** ([A2] 3.6.6.2 項) に、以下の項目を定義すること。

[機能定義]

- 1) **Trigger class 種別**  
 $\coloneqq$  **Trigger class** の種別 ([A2] 3.6.6.2 項参照 ; “**command**”, “**event**”, または “**spontaneous**” の何れか)
- 2) **Trigger class 名 (optional)**  
 $\coloneqq$  **Trigger class** の名前  
**Trigger class** 種別が **command** または **event** の場合、それぞれ、トリガとなる **Operation** (3.2.5.4 項) の **参照名** またはトリガとなる **Event class** (3.6.2.1 項) の **参照名** を、本項目に指定すること。さもなければ、本項目は指定しないこと。

### 3.4. Attribute Change Rule Definition // Attribute Change Rule 定義

#### 3.4.1. General // 一般

The rule for changes of **Attribute** values after an execution of an **Operation** is called the **Attribute Change Rule** of the **Operation** (see Section 3.4.2 in [A2]).

ある **Operation** の実行後の **Attributes** の変化の規則を、その **Operation** の **Attribute Change Rule** と呼ぶ ([A2] 3.4.2 項参照)。

The **Attribute Change Rule Definition** is the definition of how **Telecommand Issue SW** performs **Success Verifications** (Section 2.6.4.5).

An **Attribute Change Rule Definition** **shall** define zero or more **Attribute Change Rules** for a **Functional Object**.

**Attribute Change Rule 定義**は、**テレコマンド発行 SW** が **Success Verifications** (2.6.4.5 項) をどのように実施するかの定義である。

**Attribute Change Rule 定義**は、各 **Functional Object** に **Attribute Change Rules** をゼロ個以上定義すること。

### 3.4.2. Attribute Change Rule

#### 3.4.2.1. General // 一般

For an **Attribute Change Rule**, the following items **shall** be defined:

##### [Function Definition]

- 1) **Functional Object** name  
 ::= **Absolute Name** of the **Functional Object** for which the **Attribute Change Rule** is defined
- 2) **Attribute Change Rule** name  
 ::= **Local Name** of the **Attribute Change Rule**
- 3) **Attribute Change Rule Element** (1~n)  
 See Section 3.4.2.3

##### [Information Definition]

- 1) **Success Verification Check Condition** (optional) (Section 3.4.2.2)  
 ::= the condition about the timing for **Telecommand Issue SW** to check the **Engineering Values** in the **Success Verification** after issuing an **ACTION Telecommand**

各 **Attribute Change Rule** に、以下の項目を定義すること。

##### [機能定義]

- 1) **Functional Object** 名  
 ::= **Attribute Change Rule** を定義する **Functional Object** の絶対名
- 2) **Attribute Change Rule** 名  
 ::= **Attribute Change Rule** の局所名
- 3) **Attribute Change Rule Element** (1~n)

#### 3.4.2.3 項参照

##### [情報定義]

- 1) **Success Verification チェック条件** (optional) (3.4.2.2 項)  
 ::= **ACTION Telecommand** 発行後、**テレコマンド発行 SW** が **Success Verification** にて **工学値** のチェックするタイミングに関する条件

### 3.4.2.2. Success Verification Check Condition // Success Verification チェック条件

For a Success Verification Check Condition, the following items **shall** be defined:

[Information Definition]

1) Wait Time (optional)

$\coloneqq$  waiting time before starting to assess the values (see Section 2.6.4.5)

If this item is not specified, the default value defined by the **project** is used.

If the values of the **Attributes** do not change immediately after receiving an **ACTION Telecommand**, this item **should** specify the maximum time that the value might take to change.

[Note] The **Maximum Receive Times** is an appropriate parameter to specify the wait time proportional to the communication bit-rate or telemetry period.

2) Maximum Receive Times (optional)

$\coloneqq$  the maximum number of times to retrieve a new calculation result of the **Engineering Value** and redo the assessment if the values are not expected ones (due to telemetry errors or other reasons; see Section 2.6.4.5)

If this item is not specified, the default value defined by the **project** is used.

各 Success Verification チェック条件に、以下の項目を定義すること。

[情報定義]

1) 待ち時間 (optional)

$\coloneqq$  値の判定を開始するまでの待ち時間 (2.6.4.5 項参照)

本項目が指定されない場合、**project** が定義した規定値が用いられる。

本項目は、**ACTION Telecommand** を受信後、**Attributes** の値が直ちに変化しない場合、値の変化にかかり得る最大時間を指定すべきである。

[注] 通信ビットレートまたはテレメトリ周期に比例する待ち時間は、**最大再受信回数**での指定が適している。

2) 最大再受信回数 (optional)

$\coloneqq$  (テレメトリー等の理由により) 値が想定したものでなかった場合、該当 **工学値** の新たな計算結果を取得し、判定をやり直す最大回数 (2.6.4.5 項参照)

本項目が指定されない場合、**project** が定義した規定値が用いられる。

### 3.4.2.3. Attribute Change Rule Element

For an **Attribute Change Rule Element**, the following items **shall** be defined:

[Function Definition]

- 1) **Value Change Attribute** name  
 ::= Reference Name of an **Attribute** whose value changes
- 2) **Change Rule Type**  
 ::= the type of an **Attribute Change Rule Element** (see Table 3-4)
- 3) **Constant Value** (optional)  
 If the **Change Rule Type** is **SET**, the value after a change **shall** be specified for this item. Otherwise, this item **shall** not be specified.

各 **Attribute Change Rule Element** に、以下の項目を定義すること。

[機能定義]

- 1) **値変化 Attribute** 名  
 ::= 値が変化する **Attribute** の参照名
- 2) **Change Rule Type**  
 ::= **Attribute Change Rule Element** の種類 (Table 3-4 参照)
- 3) **定数値** (optional)  
 **Change Rule Type** が **SET** の場合、変化後の値を、本項目に指定すること。さもなければ、本項目は指定しないこと。

Table 3-4: Types of **Attribute Change Rule Elements**

**Attribute Change Rule Elements** の種類

Change Rule Type	Description // 説明	Note // 備考
<b>SET</b> (A := constant) (A := 定数)	The value of the <b>Attribute</b> becomes a certain constant. <b>Attribute</b> の値がある定数となる	The constant <b>shall</b> be a value that the <b>Attribute</b> can take. 定数は、 <b>Attribute</b> が取り得る値であること。
<b>INCREASE</b> (A++)	The value of the <b>Attribute</b> increases. <b>Attribute</b> の値が増加する	
<b>DECREASE</b> (A--)	The value of the <b>Attribute</b> decreases. <b>Attribute</b> の値が減少する	

### 3.5. Attribute Limit Definition // Attribute Limit 定義

#### 3.5.1. General // 一般

An [Attribute Limit](#) (see [Action Limits](#) and [Caution Limits](#) specified in Section 3.3.7.3 in [A2]) specifies the ranges of the values of an [Attribute](#) or [Pseudo Attribute](#) in order to evaluate its [Criticality Level](#).

The [Attribute Limit Definition](#) is the definition to specify [Attribute Limits](#) for [Functional Objects](#).

An [Attribute Limit Definition](#) shall define zero or more [Attribute Limits](#) for a [Functional Object](#).

The [Data Conversion Module](#) performs [Limit Checks](#) (Section 2.6.5) according to the [Attribute Limit Definition](#).

[Attribute Limit](#) ([A2] 3.3.7.3 項が定める [Action Limits](#) 及び [Caution Limits](#) 参照) は、[Attribute](#) または [Pseudo Attribute](#) の [Criticality Level](#) を評価するために、その値の範囲を定めるものである。

[Attribute Limit 定義](#)は、[Functional Objects](#) に [Attribute Limits](#) を定める定義である。

[Attribute Limit 定義](#)は、各 [Functional Object](#) に [Attribute Limits](#) をゼロ個以上定義すること。

[Attribute Limit 定義](#)に従って、[データ変換モジュール](#)は、[Limit Checks](#) (2.6.5 項) を実施する。

#### 3.5.2. Attribute Limit

##### 3.5.2.1. General // 一般

For an [Attribute Limit](#), the following items shall be defined:

[Function Definition]

1) [Functional Object](#) name

::= [Absolute Name](#) of the [Functional Object](#) for which the [Attribute Limit](#) is defined

2) [Attribute Limit](#) name

::= [Local Name](#) of the [Attribute Limit](#)

3) [Attribute Limit Element](#) (1~n)

See Section 3.5.2.2

各 [Attribute Limit](#) に、以下の項目を定義すること。

[機能定義]

1) [Functional Object](#) 名

::= [Attribute Limit](#) を定義する [Functional Object](#) の絶対名

2) [Attribute Limit](#) 名

::= [Attribute Limit](#) の局所名

3) [Attribute Limit Element](#) (1~n)

3.5.2.2 項参照

### 3.5.2.2. Attribute Limit Element

For an **Attribute Limit Element**, the following items **shall** be defined:

[Function Definition]

- 1) **Caution Limit**: Upper Limit (optional), Lower Limit (optional)

See Section 3.3.7.3 in [A2]

Not specifying an upper or lower limits means that no upper or lower limits are present, respectively.

- 2) **Action Limit**: Upper Limit (optional), Lower Limit (optional)

See Section 3.3.7.3 in [A2]

Not specifying an **Upper Limits** or **Lower Limits** means that no upper or lower limits are present, respectively.

- 3) **Check Condition**

**::=** Condition for assessing the **Attribute Limit Element** (“**always**”, “**otherwise**”, or a **Reference Name** of a **Named Condition Expression**)

If the **Attribute Limit Element** is to be always assessed, “**always**” **shall** be specified for this item. Otherwise, a **Reference Name** of a **Named Condition Expression** (Section 3.6.2.1) or “**otherwise**” **shall** be specified for this item.

各 **Attribute Limit Element** に、以下の項目を定義すること。

[機能定義]

- 1) **Caution Limit**: 上限値 (optional), 下限値 (optional)

[A2] 3.3.7.3 項参照

上限を指定しない事は上限が無い事を意味する。下限を指定しない事は下限が無い事を意味する。

- 2) **Action Limit**: 上限値 (optional), 下限値 (optional)

[A2] 3.3.7.3 項参照

**上限値**を指定しない事は上限が無い事を意味する。**下限値**を指定しない事は下限が無い事を意味する。

- 3) **チェック条件**

**::=** **Attribute Limit Element** を評価する条件 (**名前付条件式**の参照名、“**always**”，または“**otherwise**”)

**Attribute Limit Element** を常に評価する場合、**always** を本項目に指定すること。さもなければ、**名前付条件式** (3.6.2.1 項) の参照名または **otherwise** を本項目に指定すること。

If an **Attribute Limit** has only one **Attribute Limit Element**, “**always**” **shall** be specified for this item. Otherwise, a **Reference Name** of a **Named Condition Expression** **shall** be specified for this item, except for the last **Attribute Limit Element** in an **Attribute Limit**, in which case “**otherwise**” **shall** be specified for this item.

**Attribute Limit** が **Attribute Limit Element** を一つのみ有する場合、本項目に **always** を指定すること。さもなければ、本項目に **名前付条件式の参照名** を指定すること。ただし、**Attribute Limit** の最後の **Attribute Limit Element** には、本項目に、**otherwise**（その他）を指定すること。

### 3.6. Condition and Event Definition // Condition and Event 定義

#### 3.6.1. General // 一般

A classification of an event (Section 3.2.6.1) in a Functional Object is called an Event class of the Functional Object (see Section 3.5.1.1 in [A2]).

The Named Condition Expression is a named Condition Expression and is used in the definition to specify a condition.

The Condition and Event Definition is the definition to specify Event classes and conditions for Functional Objects.

A Condition and Event Definition shall define zero or more Named Condition Expressions and Event classes for a Functional Object.

ある Functional Object における event (3.2.6.1 項) の分類を、その Functional Object の Event class と呼ぶ ([A2] 3.5.1.1 項参照)。

名前付条件式は、名前の付いた条件式であり、定義の際に条件を指定するのに用いる。

Condition and Event 定義は、Functional Objects に Event classes と条件を定める定義である。

Condition and Event 定義は、各 Functional Object に、名前付条件式及び Event classes をゼロ個以上定義すること。

### 3.6.2. Named Condition Expression and Event class // 名前付条件式及び Event class

#### 3.6.2.1. General // 一般

An Event class is either a Known Condition Event class (Section 3.6.2.2) or Unknown Condition Event class (Section 3.6.2.3).

For a Named Condition Expression and Event class, the following items **shall** be defined:

[Function Definition]

1) Functional Object name

$::=$  Absolute Name of the Functional Object for which the Named Condition Expression or Event class is defined

2) Named Condition Expression/Event class name

$::=$  Local Name of the Named Condition Expression or Event class.

3) Condition-And-Event Type

$::=$  one of a Known Condition Event class, Unknown Condition Event class, and Named Condition Expression

Event class は、Known Condition Event class (3.6.2.2 項) または Unknown Condition Event class (3.6.2.3 項) の何れかである。

名前付条件式及び Event class の各々に、以下の項目を定義すること。

[機能定義]

1) Functional Object 名

$::=$  名前付条件式または Event class を定義する Functional Object の絶対名

2) 名前付条件式・Event class 名

$::=$  名前付条件式または Event class の局所名

3) Condition-And-Event 種別

$::=$  Known Condition Event class か、 Unknown Condition Event class か、名前付条件式か、の何れか

### 3.6.2.2. Known Condition Event class

A **Known Condition Event class** is a generic term for **Event classes** of **events** that occur due to some known condition.

For a **Known Condition Event class**, the following item **shall** be also defined:

[Function Definition]

1) **Trigger Condition**

$::= \text{Condition Expression}$  (Section 3.1.4)  
for the trigger of **events** in the **Event class**

**Known Condition Event class** は、何らかの既知の条件によって発生する **events** の **Event classes** の総称である。

各 **Known Condition Event class** に以下の項目も定義すること。

[機能定義]

1) **Trigger Condition**

$::= \text{Event class}$  に属する **events** のトリガの  
条件式 (3.1.4 項)

### 3.6.2.3. Unknown Condition Event class

An **Unknown Condition Event class** is a generic term for **Event classes** of **events** that occur due to some unknown condition.

No **Trigger Condition** is specified for the **Unknown Condition Event class**.

**Unknown Condition Event class** は、何らかの未知の条件によって発生する **events** の **Event classes** の総称である。

**Unknown Condition Event class** には **Trigger Condition** を指定しない。

### 3.6.2.4. Named Condition Expression // 名前付条件式

For a **Named Condition Expression**, the following item **shall** also be defined:

[Function definition]

1) **Condition Expression**

See Section 3.1.4

各 **名前付条件式** に以下の項目も定義すること。

[機能定義]

1) **条件式**

3.1.4 項参照

### 3.7. Attribute Sequence Definition // Attribute Sequence 定義

#### 3.7.1. General //一般

An Attribute Sequence defines a sequence of the fields that store the values of Attributes of Functional Objects (see Section 3.3.7 in [A3]). Attribute Sequence は、Functional Objects の Attributes の値を格納するフィールドの並びを定義する ([A3] 3.3.7 項参照)。

The Attribute Sequence Definition is the definition to specify Attribute Sequences for Functional Objects. Attribute Sequence 定義は、Functional Objects に Attribute Sequences を定める定義である。

An Attribute Sequence Definition shall define zero or more Attribute Sequences and zero or more Blocks for each of the Functional Objects.

Attribute Sequence 定義は、対象となる Functional Objects の各々に、ゼロ個以上の Attribute Sequences 及びゼロ個以上の Blocks を定義すること。

### 3.7.2. VALUE Telemetry, SET Telecommand, and GET Telecommand

The definition of an [Attribute Sequence](#) (see Section 3.7.5.1) includes the definitions of a format of [VALUE Telemetries](#) (Section 5.3 in [A3]) that carry the values of the [Attribute Sequence](#). In some cases, it also includes the definitions (see Section 3.7.5.1 3) and 4)) of a format of [GET Telecommands](#) (Section 4.4 in [A3]) and a format of [Response VALUE Telemetries](#) (Section 5.3.1 in [A3]) and/or a format of [SET Telecommands](#) (Section 4.5 in [A3]). Note that each of the [VALUE Telemetries](#) (including [Response Value Telemetries](#)) and [SET Telecommands](#) of these contains one or more sub-fields for [Attributes](#), referred to as [Telemetry Attributes](#), in the [Attribute Values](#) field. Here, the definitions of the sequence of these sub-fields uses [Attribute Sequence Field Elements \(AS Field Elements\)](#); Section 3.7.3) and might use also, if required, [Blocks](#) (Section 3.7.5.4), [switches](#) (Section 3.7.5.2), and [cases](#) (Section 3.7.5.3).

A [Block](#) is a grouping of a sequence of fields. In the definition of an [Attribute Sequence](#) or [Block](#), an [Internal Block](#) (Section 3.7.3 3)) or [Referenced Block](#) (Section 3.7.3 4)) might be specified as a sub-field. The [Internal Block](#) is defined inside the [Attribute Sequence/Block](#) definition and the [Referenced Block](#) is defined outside the [Attribute Sequence/Block](#) definition.

All definition items in this section represent [Information Definitions](#), and so the notation of [\[Information Definition\]](#) is omitted in this section.

ある [Attribute Sequence](#) の定義 (3.7.5.1 項参照) は、その [Attribute Sequence](#) の値を運ぶ [VALUE Telemetries](#) ([A3] 5.3.1 項) のフォーマットの定義を含む。また、場合により、[GET Telecommand](#) ([A3] 4.4 項) のフォーマットと [Response VALUE Telemetry](#) ([A3] 5.3.1 項) のフォーマット、及び/または、[SET Telecommands](#) ([A3] 4.5 項) のフォーマットの定義も含む (3.7.5.1 項 3) 及び 4) 項参照)。なお、これらのうち、[VALUE Telemetries](#) ([Response Value Telemetries](#) を含む) 及び [SET Telecommands](#) は、各々、[Attribute Values](#) フィールドに、[Attributes](#) ([Telemetry Attributes](#) と称する) に対するサブフィールドを一つ以上含む。ここで、これらのサブフィールドの並びの定義においては [Attribute Sequence Field Elements \(AS Field Elements\)](#); 3.7.3 項)、また、必要に応じ、[Blocks](#) (3.7.5.4 項), [switches](#) (3.7.5.2 項), 及び [cases](#) (3.7.5.3 項) を用いる。

[Block](#) はフィールドの並びをグループ化したものである。[Attribute Sequence](#) または [Block](#) の定義では、サブフィールドとして [内部 Block](#) (3.7.3 項 3)) または [参照 Block](#) (3.7.3 項 4)) を指定する事ができる。[内部 Block](#) はその [Attribute Sequence](#)・[Block](#) 定義の内部で定義され、[参照 Block](#) はその [Attribute Sequence](#) または [Block](#) 定義の外部で定義される。

以下、本項の定義項目は全て [情報定義](#) であるため、本項では [\[情報定義\]](#) の表記を省略する。



### 3.7.3. Attribute Sequence Field Element (AS Field Element)

An Attribute Sequence Field Element (AS Field Element) defines a sub-field of the Attribute Values field of a VALUE Telemetry (including Response VALUE Telemetry) or SET Telecommand. An AS Field Element is one of the following items:

- 1) Telemetry/ParameterValueSetting Attribute name (attributeRef)  
 $\coloneqq$  Reference Name of the Telemetry Attribute (\*\*1) or Value Setting Attribute (\*\*2)
 

\*\*1: Attribute whose value is notified of through a sub-field of the Attribute Values field of a VALUE Telemetry (including Response VALUE Telemetry) (see Section 3.7.2)

\*\*2: Attribute whose value is set to the value of a sub-field of the (Parameters and) Attribute Values field of an ACTION Telecommand or SET Telecommand (see Section 3.4.4.1 in [A2])
- 2) Reserved field (reserve)  
 $\coloneqq$  a sub-field that has no functional meaning
- 3) Internal Block (block)  
 See Section 3.7.5.4  
 A Block referred to as an Internal Block is allowed to be defined in the definition of the Attribute Sequence or Block.

Attribute Sequence Field Element (AS Field Element) は、VALUE Telemetry (Response VALUE Telemetry を含む) または SET Telecommand の Attribute Values フィールドのサブフィールドを定義する。AS Field Element は、以下の項目の何れかである。

- 1) テレメトリー・パラメータ値設定 Attribute 名 (attributeRef)  
 $\coloneqq$  Telemetry Attribute (\*\*1) または Value Setting Attribute (\*\*2) の 参照名

\*\*1: VALUE Telemetry (Response VALUE Telemetry を含む) の Attribute Values フィールドのサブフィールドを通じて値を通知する Attribute (3.7.2 項参照)

\*\*2: ACTION Telecommand または SET Telecommand の (Parameters and) Attribute Values フィールドのサブフィールドの値に値を設定する Attribute ([A2] 3.4.4.1 項参照)
- 2) 予約 フィールド (reserve)  
 $\coloneqq$  サブフィールドで機能的な意味を持たないもの
- 3) 内部 Block (block)  
 3.7.5.4 項参照  
Attribute Sequence や Block の定義の中で 内部 Block と称する Block を定義する事ができる。

- 4) Referenced Block name (**blockRef**)  
::= Reference Name of a Referenced Block  
An Block defined outside the definition of the Attribute Sequence or Block, which is referred to as a Referenced Block, might be specified as a sub-field by referencing its Reference Name.
- 4) 参照 Block 名 (**blockRef**)  
::= 参照 Block の参照名  
  
Attribute Sequence や Block の定義の外部で定義された Block (参照 Block と称する) を、その参照名の参照により、サブフィールドに指定する事ができる。

### 3.7.4. Functional Object

#### 3.7.4.1. General // 一般

For the **Functional Object** that is the source of **VALUE Telemetries**, which carry the values of **Attribute Sequences**, and that is the destination of **SET Telecommands** and **GET Telecommands**, the following items **shall** be defined:

1) **Functional Object name**

$::=$  **Absolute Name** of the **Functional Object**

This specification is redundant and **shall** be identical to those specified in the **Functional Object Definition**.

2) **UFORID (1~n) (\*\*1)** (Section 3.7.4.2)

$::=$  the identifier of the combination of the **Upper FOID** of the **Functional Object** (*i.e.*, the source of **VALUE Telemetries** (Section 3.7.5.1), including **Response Value Telemetries**, and the destination of **SET Telecommands** and **GET Telecommands** (see Section 3.7.5.1) all of which are specified in the definition of this **Functional Object** in the **Attribute Sequence Definition**) and the delivery **Route** of these **Telemetries** and **Telecommands**

The number of the **UFORIDs** for which no **Route** name is specified **shall** be zero or one.

**Attribute Sequences** の値を運ぶものである **VALUE Telemetries** の生成元でありかつ **SET Telecommands** 及び **GET Telecommands** の配布先である **Functional Object** に、以下の項目を定義すること。

1) **Functional Object 名**

$::=$  **Functional Object** の絶対名

この指定は冗長であり、**Functional Object 定義**で指定したものと同一であること。

2) **UFORID (1~n) (\*\*1) (3.7.4.2 項)**

$::=$  **Functional Object** (つまり、**Attribute Sequence 定義**において本 **Functional Object** の定義が規定する **VALUE Telemetries** (**Response Value Telemetries** を含む) (3.7.5.1 項) 全ての生成元、かつ **SET Telecommands** 及び **GET Telecommands** (3.7.5.1 項参照) 全ての配布先) の **Upper FOID**, 並びに、これらの **Telemetries** 及び **Telecommands** の配送 **Route** の組み合わせの識別子

**Route** 名を指定しない **UFORIDs** の数は0か1であること。

- |   |  |
|---|--|
| <p>3) <b>Lower FOID (**1)</b><br/> <math>\text{ ::= Lower FOID of the Functional Object}</math><br/>         (see Section 3.3.6 in [A3])</p> <p>4) <b>Attribute Sequence (1~n)</b><br/>         See Section 3.7.5.1</p> <p>**1: The definition in this section is redundant and the <b>UFORID</b> and <b>Lower FOID</b> <b>should</b> be identical to those specified in the <b>Functional Object Definition</b> according to [A3].</p> | <p>3) <b>Lower FOID (**1)</b><br/> <math>\text{ ::= Functional Object の Lower FOID ([A3]}</math><br/>         3.3.6 項参照)</p> <p>4) <b>Attribute Sequence (1~n)</b><br/>         3.7.5.1 項参照</p> <p>**1: [A3] に従い、本項の定義は冗長であり、<b>UFORID</b> 及び <b>Lower FOID</b> は、<b>Functional Object</b> 定義で指定したものと同一であるべきである。</p> |
|---|--|

### 3.7.4.2. UFORID

For a **UFORID**, the following items **shall** be defined:

- |  |  |
|--|--|
| <p>1) <b>Value</b><br/> <math>\text{ ::= the value of the UFORID (UFORID = Upper FOID + Route ID = Lower APID; see Section 3.3.5 in [A3])}</math></p> <p>For a <b>Functional Object</b>, the value of the <b>Upper FOID</b> embedded in any of <b>UFORIDs</b> <b>should</b> be identical according to [A3].</p> <p>The value of the <b>Route ID</b> embedded in a <b>UFORID</b> <b>should</b> be zero if no <b>Route</b> name is specified for the <b>UFORID</b>; [Rationale] for easy understanding.</p> <p>2) <b>Route name (optional)</b><br/> <math>\text{ ::= Local Name of the Route}</math></p> <p>If the <b>UFORID</b> indicates that the delivery <b>Route</b> is not specified, this item <b>shall</b> not be specified. Otherwise, this item <b>shall</b> be specified.</p> | <p>各 <b>UFORID</b> に、以下の項目を定義すること。</p> <p>1) <b>値</b><br/> <math>\text{ ::= UFORID の値 (UFORID = Upper FOID + Route ID = Lower APID; [A3]}</math><br/>         3.3.5 項参照)</p> <p>[A3] に従って、ある <b>Functional Object</b> に対して、何れの <b>UFORID</b> に埋め込まれる <b>Upper FOID</b> の値も、同一であるべきである。</p> <p>ある <b>UFORID</b> に <b>Route</b> 名を指定しない場合、その <b>UFORID</b> に埋め込まれる <b>Route ID</b> の値はゼロであるべきである；[根拠] 分かりやすくするため。</p> <p>2) <b>Route 名 (optional)</b><br/> <math>\text{ ::= Route の局所名}</math></p> <p><b>UFORID</b> が配達 <b>Route</b> を明示しない事を示す場合、本項目は指定しないこと。さもなければ、本項目を指定すること。</p> |
|--|--|

See Section 3.2.2.2 for an Example and Note.

例と注は 3.2.2.2 項参照。

### 3.7.5. Attribute Sequence and VALUE Telemetry

#### 3.7.5.1. General // 一般

Zero or more **Attribute Sequences** are defined for each of the **Functional Objects** specified according to Section 3.7.4.1.

For an **Attribute Sequence** and a **VALUE Telemetry** which reports the values of the **Attributes** that belong to the **Attribute Sequence**, the following items **shall** be defined:

- 1) **Attribute Sequence name**  
 ::= Local Name of the **Attribute Sequence**
- 2) **Attribute ID**  
 ::= the identifier of the **Attribute Sequence** in the **VALUE Telemetry** (including **Response VALUE Telemetry**), **SET Telecommand**, or **GET Telecommand** (see Section 5.3.3 in [A3])  
  
[Note] The **Attribute IDs** of **Attributes** and **Attribute Sequences** share a common ID space [Section 3.2.3.3 **Information Definition 6**].
- 3) **isGettableOrNot**  
 ::= whether a format of **GET Telecommands** and a format of **Response VALUE Telemetries** are specified for the **Attribute Sequence** or not (either “**gettable**” or “**not**”)
- 4) **isSettableOrNot**  
 ::= whether a format of **SET Telecommands** are specified for the **Attribute Sequence** or not (either “**settable**” or “**not**”)

3.7.4.1 項に従って定義した **Functional Objects** の各々に、**Attribute Sequences** をゼロ個以上定義する。

各 **Attribute Sequence** 及びその **Attribute Sequence** に属する **Attributes** の値を通知する各 **VALUE Telemetry** に、以下の項目を定義する **こと**。

- 1) **Attribute Sequence 名**  
 ::= **Attribute Sequence** の **局所名**
- 2) **Attribute ID**  
 ::= **VALUE Telemetry** (**Response VALUE Telemetry** を含む), **SET Telecommand**, または **GET Telecommand** の **Attribute Sequence** の識別子 ([A3] 5.3.3 項参照)
- 3) **GET 可能か否か**  
 ::= **Attribute Sequence** に対して、**GET Telecommands** のフォーマットと **Response VALUE Telemetries** のフォーマットを規定するか否か (“**gettable**” または “**not**” の何れか)
- 4) **SET 可能か否か**  
 ::= **Attribute Sequence** に対して、**SET Telecommands** のフォーマットを規定するか否か (“**settable**” または “**not**” の何れか)

[注] **Attributes** と **Attribute Sequences** の **Attribute IDs** は、共通の ID 空間を共有する [3.2.3.3 項情報定義 6)]。

- 5) Generation timing  
 $\coloneqq$  the generation timing of the **VALUE Telemetries** (excluding the Response **VALUE Telemetries**)  
 [Note] No format is specified.
- 6) **Upper APID**  
 $\coloneqq$  the entities that are allowed to monitor the **VALUE Telemetry** (including the Response **VALUE Telemetry**) (see Section 5.2.3 in [A4]; either “**any**” (meaning “monitored by **Controllers** and **Monitors**”) or “**monitor\_only**” (“monitored by **Monitors** only”))
- 7) **AS Field Element** (0~n)  
 See Section 3.7.3  
 The order of the definitions of the **AS Field Elements**, **switches**, and **cases** **shall** be the same as the order of the sub-fields of the **Attribute Values** field of the **VALUE Telemetry** or **SET Telecommand**, including the nested structure that use the **Blocks**, **switches**, and/or **cases**.
- 8) **switch** (0~n) (Section 3.7.5.2)  
 $\coloneqq$  field sequences, one of which is valid according to the value of a specified **Attribute** for each **VALUE Telemetry** (deprecated because it does not comply with the **SMCP**)
- 5) 生成タイミング  
 $\coloneqq$  **VALUE Telemetries** (Response **VALUE Telemetries** は含まない) の生成タイミング  
 [注] 書式は規定されていない。
- 6) **Upper APID**  
 $\coloneqq$  **VALUE Telemetry** (Response **VALUE Telemetry** を含む) の監視を許容する構成要素 ([A4] 5.2.3 項参照; “**any**”: **Controllers** と **Monitors** によって監視される, または “**monitor\_only**”: **Monitors** のみによって監視される、の何れか)
- 7) **AS Field Element** (0~n)  
 3.7.3 項参照  
**AS Field Elements**, **switches**, 及び **cases** の定義順は、**Blocks**, **switches**, 及び・または **cases** を用いた入れ子構造も含めて、**VALUE Telemetry** または **SET Telecommand** の **Attribute Values** フィールドのサブフィールドの並び順と同一であること。
- 8) **switch** (0~n) (3.7.5.2 項)  
 $\coloneqq$  指定した **Attribute** の値に従って **VALUE Telemetry** 毎に何れか一つの並びが有効となる、フィールド並びの組 (**SMCP** を逸脱するため非推奨)

### 3.7.5.2. switch

For a **switch**, the following items **shall** be defined:

- |   |   |
|---|---|
| 1) Referenced <b>Attribute</b> name<br>::= <b>Reference Name</b> of an <b>Attribute</b> used for the case identification  | 1) 参照 <b>Attribute</b> 名<br>::= 場合分けに使用する <b>Attribute</b> の <b>参照名</b>   |
| 2) <b>case</b> (2~n) (Section 3.7.5.3)<br>::= a field sequence<br><br>The sum of the bit lengths of the fields of any of the <b>cases</b> in a <b>switch</b> <b>shall</b> be the same except for the <b>cases</b> in the <b>switch</b> that is located at the end of an <b>Attribute Sequence</b> . | 2) <b>case</b> (2~n) (3.7.5.3 項)<br>::= フィールド並び<br><br>ある <b>switch</b> に含まれる <b>cases</b> は、何れもフィールドのビット長の総和が、 <b>Attribute Sequence</b> の末尾にある <b>switch</b> の <b>cases</b> を除き、同一である <b>こと</b> 。 |

### 3.7.5.3. case

For a **case**, the following items **shall** be defined:

- |   |   |
|---|---|
| 1) Condition <b>Attribute</b> Value<br>::= <b>Attribute</b> value that identifies the <b>case</b> | 1) 条件 <b>Attribute</b> 値<br>::= <b>case</b> を示す <b>Attribute</b> の値 |
| 2) <b>AS Field Element</b> (0~n)<br><br>See Section 3.7.3   | 2) <b>AS Field Element</b> (0~n)<br><br>3.7.3 項参照                   |
- [Note] **switches** are not allowed to be nested.
- [注] **switches** の入れ子は許容されていない。

### 3.7.5.4. Internal Block // 内部 Block

For an **Internal Block**, the following items **shall** be defined:

- |  |   |
|--|---|
| 1) <b>Functional Object</b> name<br>::= <b>Absolute Name</b> of the <b>Functional Object</b> for which the <b>Block</b> is defined   | 1) <b>Functional Object</b> 名<br>::= <b>Block</b> を定義する <b>Functional Object</b> の絶対名   |
| 2) <b>Block</b> name (optional)<br>::= <b>Local Name</b> of the <b>Block</b>   | 2) <b>Block</b> 名 (optional)<br>::= <b>Block</b> の局所名   |
| 3) <b>Bit-Length</b>   | 3) <b>Bit-Length</b>  |
| 4) <b>AS Field Element</b> (0~n)<br><br>See Section 3.7.3<br><br><b>Blocks</b> are allowed to be nested. However, the nesting level of <b>Blocks</b> <b>shall</b> be no more than three including the references of the external <b>Blocks</b> . | 4) <b>AS Field Element</b> (0~n)<br><br>3.7.3 項参照<br><br><b>Blocks</b> の入れ子も可能である。ただし、 <b>Blocks</b> の入れ子は外部の <b>Blocks</b> の参照も含めて三階層以内である <b>こと</b> 。 |

### 3.7.5.5. Telemetry/ParameterValueSetting Attribute

#### テレメトリ・パラメータ値設定 Attribute

The following item **may** be specified for a **Telemetry/ParameterValueSetting Attribute**. Note that specifying it is redundant. If the following item is specified, it **shall** be identical to that specified in the **Functional Object Definition** [Section 3.2.3.3 Information Definition 3)].

- |                            |                            |
|----------------------------|----------------------------|
| 1) <b>Bit-Length</b> (0~2) | 1) <b>Bit-Length</b> (0~2) |
|----------------------------|----------------------------|

各 **テレメトリ・パラメータ値設定 Attribute** に以下の項目を定めて**良い**。ただし、この指定は冗長である。指定する場合、**Functional Object 定義** [3.2.3.3 項情報定義 3)] で指定したものと同一である**こと**。

### 3.7.5.6. Reserved field and Referenced Block // 予約フィールド及び参照 Block

For each of **reserved** fields and **Referenced Blocks**, the following item **shall** be defined:

- |                      |                      |
|----------------------|----------------------|
| 1) <b>Bit-Length</b> | 1) <b>Bit-Length</b> |
|----------------------|----------------------|

**予約フィールド及び参照 Blocks** の各々に、以下の項目を定義すること。

- |                      |
|----------------------|
| 1) <b>Bit-Length</b> |
|----------------------|

## 3.8. Conversion Definition // Conversion 定義

### 3.8.1. General // 一般

The [Conversion Definition](#) is the definition for the [Data Conversion Module](#) to perform [Engineering Unit Conversions](#).

A [Conversion Definition](#) **shall** define zero or more [Engineering Unit Conversions](#) (Section 2.5.5.2) for a [Functional Object](#).

All definition items in this section represent [Information Definitions](#) and so the notation of [[Information Definition](#)] is omitted in this section.

[Conversion 定義](#)は、[データ変換モジュール](#)が [工学値変換](#)を実施するための定義である。

[Conversion 定義](#)は、各 [Functional Object](#) に、[工学値変換](#) (2.5.5.2 項) をゼロ個以上定義すること。

以下、本項の定義項目は全て[情報定義](#)であるため、本項では [情報定義] の表記を省略する。

### 3.8.2. Engineering Unit Conversion // 工学値変換

#### 3.8.2.1. General // 一般

For an Engineering Unit Conversion, the following items **shall** be defined:

- |  |   |
|--|---|
| 1) Functional Object name<br>::= Absolute Name of the Functional Object for which the Engineering Unit Conversion is defined   | 1) Functional Object 名<br>::= 工学値変換を定義する Functional Object の絶対名                       |
| 2) Engineering Unit Conversion name<br>::= Local Name of the Engineering Unit Conversion   | 2) 工学値変換名<br>::= 工学値変換の局所名  |
| 3) <u>Number of Significant Digits</u> (optional)<br>::= the number of the significant digits of the Engineering Value<br><br>If this item is not specified, the default value defined by the project is used. | 3) <u>有効桁数</u> (optional)<br>::= 工学値の有効桁数<br><br>本項目が指定されない場合、project が定義した規定値が用いられる。 |
| 4) <u>Unit</u> (optional) <sup>29</sup><br>::= the unit of the Engineering Value   | 4) <u>単位</u> (optional) <sup>30</sup><br>::= 工学値の単位                                   |

---

<sup>29</sup> This item is mandatory in the current implementation as of March 2023.

<sup>30</sup> 2023 年 3 月現在の実装では本項目は必須である。

- 5) Engineering Unit Conversion Element (1~n) (Section 3.8.2.2)
- $::=$  An element to define an Engineering Unit Conversion: either an Engineering Unit Conversion Element with an Arithmetic Expression with a condition (see Section 3.8.2.3) (hereafter, a Conditional Engineering Unit Conversion Element) or an Engineering Unit Conversion Element with an Arithmetic Expression without a condition (see Section 3.8.2.3) (hereafter, a Non-Conditional Engineering Unit Conversion Element)
- The Engineering Unit Conversion Elements of an Engineering Unit Conversion **shall** consist of a Non-Conditional Engineering Unit Conversion Elements only or Conditional Engineering Unit Conversion Elements only.
- If multiple Conditional Engineering Unit Conversion Elements are specified for an Engineering Unit Conversion, the conditions of the Engineering Unit Conversion Elements **shall** be mutually exclusive.
- [Note 1] If multiple Non-Conditional Engineering Unit Conversion Elements are specified for an Engineering Unit Conversion, the Data Conversion Module evaluates the Engineering Unit Conversion Elements sequentially.
- [Note 2] The result of Engineering Unit Conversion is undefined if the constraints is not satisfied.
- 5) 工学値変換要素 (1~n) (3.8.2.2 項)
- $::=$  工学値変換を定義するための要素: 条件付きの演算式 (3.8.2.3 項参照) を持つもの (以下、条件付工学値変換要素) か、条件無しの演算式 (3.8.2.3 項参照) を持つもの (以下、条件無工学値変換要素) かの何れか
- ある工学値変換の工学値変換要素は、条件付工学値変換要素のみか、条件無工学値変換要素のみからなること。
- ある工学値変換に条件付工学値変換要素を複数定める場合、これらの工学値変換要素の条件は相互に排他的であること。
- [注 1] ある工学値変換に条件無工学値変換要素を複数定める場合、データ変換モジュールは、これらの工学値変換要素を順次評価する。
- [注 2] 制約が満たされない場合、工学単位変換の結果は未定義である。

### 3.8.2.2. Engineering Unit Conversion Element // 工学値変換要素

For an Engineering Unit Conversion Element, the following items **shall** be defined:

1) Arithmetic Expression

See Section 3.8.2.3

2) Conversion Effective-Condition name  
(optional)

Unless the type of the Arithmetic Expression is conditional, this item **shall** not be specified. Otherwise, a Reference Name of the Conversion Effective-Condition (the Condition Expression under which the Engineering Unit Conversion Elements is valid) **shall** be specified for this item, except for the last Engineering Unit Conversion Element in the Engineering Unit Conversion, in which case “**otherwise**” **shall** be specified for this item.

各工学値変換要素に、以下の項目を定義すること。

1) 演算式

3.8.2.3 項参照

2) 変換有効条件名 (optional)

演算式の種類が条件付でない場合、本項目は指定しないこと。さもなければ、本項目に、変換有効条件（工学値変換要素が有効となる名前付条件式）の参照名を指定すること。ただし、工学値変換の最後の工学値変換要素には、“otherwise”（その他）を、本項目に指定すること。

### 3.8.2.3. Arithmetic Expression // 演算式

An Arithmetic Expression **shall** be one of those shown in Table 3-5. There is a constraint requiring that neither polynomial nor conditional polynomial **shall** be used for the Pseudo Attribute.

演算式は、Table 3-5 に示すものの何れかであること。ただし、Pseudo Attribute には、polynomial 及び conditionalPolynomial は用いないことという制約がある。

Table 3-5: Types of Arithmetic Expressions // 演算式の種類

Types of Arithmetic Expressions // 演算式の種類	Explanations 説明
<b>polynomial</b> polynomial conversion 多項式変換	$y = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ Conversion by 5th order polynomial 5次多項式による変換
<b>conditionalPolynomial</b> conditional polynomial conversion 条件付多項式変換	Polynomial conversion is performed only if the condition is true 条件が真の場合にのみ多項式変換を実施
<b>expression</b> user-defined arithmetic expression conversion ユーザ定義演算式変換	Conversion according to arithmetic formulae of constants, mathematical operators, and built-in functions (including multiple four-basic arithmetic operations in combination with parentheses). 定数、数学的演算子、組み込み関数の演算式に従った変換（複数の四則演算とカッコの組合せを含む）。 Example : A+B, A-B, A*B, A+B+C, (A+B) *C (where each of A and B denotes an <a href="#">Attribute</a> name or <a href="#">Parameter</a> name). 例 : A+B, A-B, A*B, A+B+C, (A+B) *C (ここで A や B は <a href="#">Attribute</a> 名または <a href="#">Parameter</a> 名を表すものとする。)
<b>conditionalExpression</b> conditional user-defined arithmetic expression conversion 条件付ユーザ定義演算式変換	User-defined arithmetic expression conversion is performed only if the condition is true 条件が真の場合にのみユーザ定義演算式変換を実施
<b>function</b> user-defined function conversion ユーザ定義関数変換	Call a user-defined function with the referenced <a href="#">Attributes</a> and constant values as arguments. 参照する <a href="#">Attributes</a> 及び定数値を引数としユーザ定義関数を呼び出す。

Constants used in user-defined arithmetic conversions, mathematical operators, and built-in functions **shall** be as follows.

Constant: "real value" | "integer value"

**x** : Raw Data Value, which is obtained from a Binary Sequence Value as a result of Numerical Decoding (not available for the Pseudo Attribute)

**y** : Engineering Value (obtained in the immediate previous Engineering Unit Conversion)

Here, neither **x** nor **y** **shall** be used for the Pseudo Attribute.

Mathematical operator: + | - | \* | / | % | ^

Built-in function:= sin() | cos() | tan() | asin() | acos() | atan() | atan2() | sinh() | cosh() | tanh() | asinh() | acosh() | atanh() | log() | log10() | exp() | sqrt() | abs() | floor() | ceil() | max() | min()

ユーザ定義演算式変換で用いる定数、数学的演算子及び組み込み関数は、以下とすること。

定数：「実数値」 | 「整数値」

**x** : (数値デコードの結果バイナリ列値から得られたるものである) 生データ値 (Pseudo Attribute に対しては使用不可)

**y** : (直前の工学値変換で得られた) 工学値

ここで、Pseudo Attribute には、**x** も **y** も用いないこと。

数学的演算子: + | - | \* | / | % | ^

組み込み関数: = sin() | cos() | tan() | asin() | acos() | atan() | atan2() | sinh() | cosh() | tanh() | asinh() | acosh() | atanh() | log() | log10() | exp() | sqrt() | abs() | floor() | ceil() | max() | min()

### 3.9. Enumeration Definition // Enumeration 定義

#### 3.9.1. General // 一般

The **Enumeration Definition** is the definition for the **Data Conversion Module** to perform **String Conversions** (Section 2.5.6.2).

An **Enumeration Definition** **shall** define zero or more **Valid Value Sets** (Section 3.3.6.2 in [A2]) for a **Functional Object**.

**Enumeration 定義**は、データ変換モジュールが**文字列変換**（2.5.6.2 項）を実施するための定義である。

**Enumeration 定義**は、各 **Functional Object** に、**Valid Value Sets** ([A2] 3.3.6.2 項) をゼロ個以上定義すること。

#### 3.9.2. Valid Value Set

##### 3.9.2.1. General // 一般

For a **Valid Value Set**, the following items **shall** be defined:

[Function Definition]

- 1) **Functional Object** name  
::= Absolute Name of the Functional Object for which the Valid Value Set is defined
- 2) **Valid Value Set** name  
::= Local Name of the Valid Value Set
- 3) **Enumeration Element** (1~n)  
See Section 3.9.2.2

各 **Valid Value Set** に、以下の項目を定義すること。

[機能定義]

- 1) **Functional Object** 名  
::= **Valid Value Set** を定める **Functional Object** の絶対名
- 2) **Valid Value Set** 名  
::= **Valid Value Set** の局所名
- 3) **Enumeration Element** (1~n)

3.9.2.2 項参照

### 3.9.2.2. Enumeration Element

For each **Enumeration Element** in the **Valid Value Set**, the following items **shall** be defined:

[Function Definition]

1) **Enumerative Name**

$\coloneqq$  **Local Name** that identifies the value (*i.e.*, a character string)

2) **Criticality Level** (optional)

$\coloneqq$  **Criticality Level** of the value (see Section 3.3.6.3 in [A2]; one of **Normal**, **Caution**, **Action**, and the color names (“blue”, “magenta”, “cyan”, and “white”))

If this item is not specified, **white** is deemed to be specified.

[Note] **Telemetry Monitor SW** displays the character string obtained in a **String Conversion** in the color according to the **Criticality Level** (see Section 2.6.6.3). Note that **Normal**, **Caution**, and **Action** are mapped to colors “white”, “yellow”, and “red”, respectively.

[Information Definition]

1) **Raw Data Value**

**Valid Value Set** 内の各 **Enumeration Element** に、以下の項目を定義すること。

[機能定義]

1) **Enumerative Name**

$\coloneqq$  値を識別する**局所名** (つまり、文字列)

2) **Criticality Level** (optional)

$\coloneqq$  値の **Criticality Level** ([A2] 3.3.6.3 項参照; **Normal**, **Caution**, **Action**, または色名 (“blue”, “magenta”, “cyan”, または “white”) の何れか)

本項目が指定されない場合、**white** が指定されたとみなされる。

[注] **テlemetry 監視 SW** は、文字列変換で得られた文字列を、**Criticality Level** に従う色で画面に表示する (2.6.6.3 項参照)。ここで、**Normal**, **Caution** 及び **Action** は、それぞれ、“white”, “yellow” 及び “red” にマッピングされる。

[情報定義]

1) **生データ値**

## 4. Memory Functional Object Definition

### Memory Functional Object 定義

#### 4.1. General // 一般

The [Memory Functional Object Definition](#) is the definition that specifies the design parameters of [Memory Functional Objects](#) and the configuration parameters related to the function of [Telecommand Issue SW](#) for the [Memory Functional Objects](#) (see Section 2.6.7).

[Memory Functional Object 定義](#)は、[Memory Functional Objects](#) の設計パラメータとそれらの [Memory Functional Objects](#) に関する [テレコマンド発行 SW](#) の機能（2.6.7 項参照）に関する設定パラメータを指定するための定義である。

## 4.2. Memory Functional Object

For a **Memory Functional Object**, the following items **shall** be defined:

- 1) **Memory Functional Object** name  
 ::= Absolute Name of the Memory Functional Object
- 2) **FirstAddress**
- 3) **LastAddress**

[Note] See Section 5.2 in [A2] for the definitions of the above-listed design parameters 2) and 3).

- 4) **Collation Range** (a pair of Start Address and End Address) (1~n)

::= the address range to be collated by **Telecommand Issue SW**

- 5) Whether **Memory Management** is performed or not  
 ::= whether **Telecommand Issue SW** manages the memory contents of the **Memory Functional Object** (**Memory Management**; Section 2.6.7) or not

- 6) **Route Definition** (1~n)

See Section 4.3.1

- 7) **Check Sum Definition** (optional)

See Section 4.4

各 **Memory Functional Object** に、以下の項目を定義する **こと**。

- 1) **Memory Functional Object** 名  
 ::= **MemoryFunctional Object** の絶対名

- 2) **FirstAddress**
- 3) **LastAddress**

[注] 上記の設計パラメータ 2), 3)の定義は [A2] 5.2 項参照。

- 4) **照合範囲** (Start Address と End Address との組) (1~n)

::= **テレコマンド発行 SW** による照合対象のアドレス範囲

- 5) **メモリ管理実施有無**

::= **テレコマンド発行 SW** が **Memory Functional Object** のメモリ内容を管理（**メモリ管理**; 2.6.7 項）するか否か

- 6) **経路定義** (1~n)

4.3.1 項参照

- 7) **Check Sum 定義** (optional)

4.4 項参照

## 4.3. Route Definition // 経路定義

### 4.3.1. General // 一般

- For a [Route Definition](#), the following items **shall** be defined:
- 1) [Route name](#)  
 $\text{ ::= } \text{the name of the route ("normal" if the default one is specified)}$
  - 2) [Telemetry Definition](#) (optional)  
 See Section 4.3.3 (This item defines a format of [MEMORY DUMP Telemetries](#).)  
 [Note] A Memory Functional Object generates [MEMORY DUMP Telemetries](#) for a [MEMORY DUMP Telecommand](#) according to [SMCP](#). Thus, this item is specified if a [READ Command Definition](#) is specified in the [Telecommand Definition](#).
  - 3) [Telecommand Definition](#) (optional)  
 See Section 4.3.2.1 (This item defines a format of [MEMORY DUMP Telecommands](#) and/or a format of [MEMORY LOAD Telecommands](#).)
- 各経路定義に以下を定義すること。
- 1) [Route 名](#)  
 $\text{ ::= } \text{経路の名前 (デフォルトを指定する場合は "normal")}$
  - 2) [Telemetry 定義](#) (optional)  
 4.3.3 項参照 (本項目は、[MEMORY DUMP Telemetries](#) のフォーマットを定義する。)  
 [注] Memory Functional Object は、[SMCP](#) に従い [MEMORY DUMP Telecommand](#) に対して [MEMORY DUMP Telemetries](#) を生成する。そこで、[Telecommand 定義](#) で [READ Command 定義](#) を指定する場合は、本項目を指定する事になる。
  - 3) [Telecommand 定義](#) (optional)  
 4.3.2.1 項参照(本項目は、[MEMORY DUMP Telecommands](#) のフォーマット及び/または [MEMORY LOAD Telecommands](#) のフォーマットを定義する。)

### 4.3.2. Telecommand Definition // Telecommand 定義

#### 4.3.2.1. General // 一般

A **Telecommand Definition** defines a format of **MEMORY DUMP Telecommands** and/or a format of **MEMORY LOAD Telecommands**.

For a **Telecommand Definition**, the following items **shall** be defined for a format of telecommands (**MEMORY LOAD Telecommands** and a format of **MEMORY DUMP Telecommands**):

- 1) **Telecommand MAP ID** (optional)  
 $\text{ ::= MAP ID of the TC Transfer Frames that carry the telecommands}$   
 This item **shall** be specified if the **MAP Service** is applied to the telecommands. Otherwise, this item **shall** not be specified.
- 2) **Telecommand APID**  
 $\text{ ::= APID of the Packets that carry the telecommands}$   
 The upper 3 bits of the **APID (Upper APID)** is the one specified in Section 5.2.3 in [A4] (*i.e.*, 000b).  
 The lower 8 bits of the **APID (Lower APID)** is the **UFORID** (*i.e.*, the identifier of the combination of the **Upper FOID** of the **Memory Functional Object** and the delivery **Route** of the telecommands; see Section 3.2.2 in [A3]).

**Telecommand 定義**は、**MEMORY DUMP Telecommands** のフォーマット及び/または **MEMORY LOAD Telecommands** のフォーマットを定義する。

各 **Telecommand 定義**では、テレコマンド (**MEMORY LOAD Telecommands** のフォーマット及び **MEMORY DUMP Telecommands**) のフォーマットに対して、以下を定義すること。

- 1) **Telecommand MAP ID** (optional)  
 $\text{ ::= 同テレコマンドを運ぶ TC Transfer Frames の MAP ID}$   
 同テレコマンドに **MAP Service** を適用する場合、本項目を指定すること。さもなければ、本項目を指定しないこと。
- 2) **Telecommand APID**  
 $\text{ ::= 同テレコマンドを運ぶ Packets の APID}$

**APID** の上位 3bits (**Upper APID**) は、[A4] 5.2.3 項が定めたものである (すなわち、000b)。

**APID** の下位 8bits (**Lower APID**) は、**UFORID** (すなわち同 **Memory Functional Object** の **Upper FOID**, 並びに、同テレコマンドの配送 **Route** の組み合わせの識別子; [A3] 3.2.2 項参照) である。

For a **Memory Functional Object**, the value of the **Upper FOID** embedded in any of **UFORIDs** **should** be identical regardless of whether they are for a telecommand or telemetry and regardless of their routes, according to [A3].

3) **Lower FOID**

::= **Lower FOID** of the telecommands (see Section 3.3.6 in [A3])

The **Lower FOID** of any of **MEMORY LOAD Telecommands**, **MEMORY DUMP Telecommands**, and **MEMORY DUMP Telemetries** for a **Memory Functional Object** **should** be identical regardless of their routes according to [A3].

4) **WRITE Command Definition** (optional)

See Section 4.3.2.2 (This item defines a format of **MEMORY LOAD Telecommands**.)

5) **READ Command Definition** (optional)

See Section 4.3.2.3 (This item defines a format of **MEMORY DUMP Telecommands**.)

[A3] に従い、ある **MemoryFunctional Object** に対して、何れの **UFORID** に埋め込まれる **Upper FOID** の値も、それらがテレコマンドかテlemetryかに対応するものか及びその経路によらず、同一であるべきである。

3) **Lower FOID**

::= 同テレコマンドの **Lower FOID** ([A3] 3.3.6 項参照)

ある **Memory Functional Object** に対する **MEMORY LOAD Telecommands**, **MEMORY DUMP Telecommands**, 及び **MEMORY DUMP Telemetries** の何れの **Lower FOID** も、[A3] に従い、その経路によらず、同一であるべきである。

4) **WRITE Command 定義** (optional)

4.3.2.2 項参照(本項目は、**MEMORY LOAD Telecommands** のフォーマットを定義する。)

5) **READ Command 定義** (optional)

4.3.2.3 項参照(本項目は、**MEMORY DUMP Telecommands** のフォーマットを定義する。)

In the following, the **Absolute Name** of a **Command Name** **should** be a combination of the **Absolute Name** of the **Memory Functional Object**, the **Local Name** of the **Route** (only if multiple **Routes** exist), and the **Local Name** of the **Command** in this order connected with a period sign in between (see Section 5.4).

以下で、各 **Command 名** の**絶対名**は、**Memory Functional Object** の**絶対名**、**Route** の**局所名** (**Routes** が複数存在する場合のみ)、及び、その **Command** の**局所名**を、この順に、それぞれピリオドを挟んで接続したものであるべきである (5.4 項参照)。

#### 4.3.2.2. WRITE Command Definition (MEMORY LOAD Telecommand)

##### WRITE Command 定義 (MEMORY LOAD Telecommand)

A Telecommand Definition defines a format of MEMORY LOAD Telecommands.

For a WRITE Command Definition, the following items **shall** be defined:

1) **WRITE Command name**

::= the name of a WRITE Command (*i.e.*, the Telecommand Name of the MEMORY LOAD Telecommands)

The Local Name of a WRITE Command **should** be WRITE; [Rationale] for easy understanding.

2) **AlignmentLength**

See Section 5.2 in [A2]

3) **MaximumUploadLength**

See Section 5.2 in [A2]

4) **MEMORY LOAD Telecommand ACK Response**

::= whether the Memory Functional Object sends an ACK Telemetry or not in response to a MEMORY LOAD Telecommand that requests an ACK Telemetry

5) **LOAD Command name (optional)**

::= the name of a LOAD Command

This item **shall** be specified if Memory Management is performed on the Memory Functional Object. Otherwise, it **shall** not be specified.

The Local Name of a LOAD Command **should** be LOAD; [Rationale] for easy understanding.

Telecommand 定義は、MEMORY LOAD Telecommands のフォーマットを定義する。

各 WRITE Command 定義に以下を定義すること。

1) **WRITE Command 名**

::= WRITE Command の名前 (つまり、MEMORY LOAD Telecommands の Telecommand 名)

WRITE Command の局所名は、WRITE であるべきである; [根拠] 分かりやすくするため。

2) **AlignmentLength**

[A2] 5.2 項を参照

3) **MaximumUploadLength**

[A2] 5.2 項を参照

4) **MEMORY LOAD Telecommand ACK 応答**

::= Memory Functional Object が、ACK Telemetry を要求する MEMORY LOAD Telecommand に対する応答として、ACK Telemetry を送出するか否か

5) **LOAD Command 名 (optional)**

::= LOAD Command の名前

本項目は、Memory Functional Object にメモリ管理を実施する場合に指定すること。さもなければ指定しないこと。

LOAD Command の局所名は、LOAD であるべきである; [根拠] 分かりやすくするため。

6) Optimization

::= whether or not Telecommand Issue SW allows Optimization (Section 2.6.7) to be performed when writing memory (valid only when Memory Management is performed on the Memory Functional Object and when the WRITE Command Definition is specified in the Telecommand Definition)

6) 最適化可否

::= テレコマンド発行 SW がメモリ書き込みの際に最適化 (2.6.7 項項参照) の実施を許可するか否か (Memory Functional Object にメモリ管理を実施、かつ、Telecommand 定義で WRITE Command 定義が指定された場合にのみ有効)

#### 4.3.2.3. READ Command Definition (MEMORY DUMP Telecommand)

##### READ Command 定義 (MEMORY DUMP Telecommand)

A Telecommand Definition defines a format of **MEMORY DUMP Telecommands**.

For a **READ Command Definition**, the following items **shall** be defined:

1) **READ Command name**

::= the name of a **READ Command** (*i.e.*, the **Telecommand Name** of the **MEMORY DUMP Telecommands**)

The **Local Name** of a **READ Command** **should** be **READ**; [Rationale] for easy understanding.

2) **MEMORY DUMP Telecommand ACK Response**

::= whether the **Memory Functional Object** sends an **ACK Telemetry** or not in response to a **MEMORY DUMP Telecommand** that requests an **ACK Telemetry**

3) **DUMP Command name**

::= the name of a **DUMP Command**

The **Local Name** of a **DUMP Command** **should** be **DUMP**; [Rationale] for easy understanding.

**Telecommand 定義** は、**MEMORY DUMP Telecommands** のフォーマットを定義する。

各 **READ Command 定義** に以下を定義すること。

1) **READ Command 名**

::= **READ Command** の名前 (つまり、**MEMORY DUMP Telecommands** の **Telecommand 名**)

**READ Command** の **局所名** は、**READ** であるべきである; [根拠] 分かりやすくするため。

2) **MEMORY DUMP Telecommand ACK 応答**

::= **Memory Functional Object** が、**ACK Telemetry** 要求する **MEMORY DUMP Telecommand** に対する応答として、**ACK Telemetry** を送出するか否か

3) **DUMP Command 名**

::= **DUMP Command** の名前

**DUMP Command** の **局所名** は、**DUMP** であるべきである; [根拠] 分かりやすくするため。

#### 4.3.3. Telemetry Definition (MEMORY DUMP Telemetry)

##### Telemetry 定義 (MEMORY DUMP Telemetry)

A Telemetry Definition defines a format of MEMORY DUMP Telemetries.

For a Telemetry Definition, the following items **shall** be defined for a format of telemetries (MEMORY DUMP Telemetries):

###### 1) Telemetry APID

::= APID of the Packets that carry the telemetries

The upper 3bits of the APID (Upper APID) is the one specified in Section 5.2.3 in [A4] (*i.e.*, either 010b or 011b).

The lower 8bits of the APID (Lower APID) is the UFORID (*i.e.*, the identifier of the combination of the Upper FOID of the Memory Functional Object and the delivery Route of the telemetries; see Section 3.2.2 in [A3]).

For a Memory Functional Object, the value of the Upper FOID embedded in any of UFORIDs **should** be identical regardless of whether they are for a telecommand or telemetry and regardless of their routes, according to [A3].

###### 2) Lower FOID

::= Lower FOID of the telemetries (see Section 3.3.6 in [A3])

The Lower FOID of any of MEMORY LOAD Telecommands, MEMORY DUMP Telecommands, and MEMORY DUMP Telemetries for a Memory Functional Object **should** be identical regardless of their routes according to [A3].

Telemetry 定義は、MEMORY DUMP Telemetries のフォーマットを定義する。

各 Telemetry 定義では、テレメトリ (MEMORY DUMP Telemetries) のフォーマットに対して、以下を定義すること。

###### 1) Telemetry APID

::= 同テレメトリを運ぶ Packets の APID

APID の上位 3bits (Upper APID) は、[A4] 5.2.3 項が定めたものである (すなわち、010b か 011b の何れか)。

APID の下位 8bits (Lower APID) は、UFORID (すなわち同 Memory Functional Object の Upper FOID, 並びに、同テレメトリの配送 Route の組み合わせの識別子; [A3] 3.2.2 項参照) である。

[A3] に従い、ある Memory Functional Object に対して、何れの UFORID に埋め込まれる Upper FOID の値も、それらがテレコマンドかテレメトリかに対応するものか及びその経路によらず、同一であるべきである。

###### 2) Lower FOID

::= 同テレメトリの Lower FOID ([A3] 3.3.6 項参照)

ある Memory Functional Object に対する MEMORY LOAD Telecommands, MEMORY DUMP Telecommands, 及び MEMORY DUMP Telemetries の何れの Lower FOID も、[A3] に従い、その経路によらず、同一であるべきである。



#### 4.4. Check Sum Definition // Check Sum 定義

For a **Check Sum Definition**, the following items **shall** be defined: 各 **Check Sum 定義**では、以下を定義する**こと**。

- |  |  |
|--|--|
| 1) <b>Check Sum Start Address</b><br>:= the start address of the memory area for which the check sum is calculated | 1) <b>Check Sum Start Address</b><br>:= チェックサムを計算するメモリ領域の開始アドレス              |
| 2) <b>Check Sum Last Address</b><br>:= the last address of the memory area for which the check sum is calculated   | 2) <b>Check Sum Last Address</b><br>:= チェックサムを計算するメモリ領域の最終アドレス               |
| 3) <b>Check Sum Value Address</b><br>:= the address of the memory area in which the check sum is stored            | 3) <b>Check Sum Value Address</b><br>:= チェックサムを格納するメモリ領域のアドレス                |
| 4) <b>Check Sum Algorithm</b><br>:= the algorithm to calculate the check sum (either “ADD32” or “XOR”)             | 4) <b>Check Sum Algorithm</b><br>:= チェックサムを計算するアルゴリズム (“ADD32” か “XOR” の何れか) |

## 5. Naming/ID assignment rules // 命名・ID 付与規則

### 5.1. General // 一般

This chapter specifies the rules for defining names and IDs.

本章では、名前と ID を定義するための規則を定める。

### 5.2. Rules for Defining Names // 名前を定義する規則

A name that is unique within an entity (such as a [Functional Object](#), [Operation](#), [Alert](#), and [State Machine](#)) is referred to as a [Local Name](#). A [Local Name](#) **shall** begin with an alphabetic character, consist only of alphanumeric characters and underscores, and be a string of 16 or less characters. They are case-insensitive.

ある物 ([Functional Object](#), [Operation](#), [Alert](#), [State Machine](#) 等) の中で一意になるように付けた名前を局所名と称する。局所名は、英字で始まり、英数字とアンダースコアのみで構成される 16 字以内の文字列である**こと**。英字の大文字と小文字は区別しないものとする。

When referring from a [Functional Object](#) to an entity defined within the [Functional Object](#) (and its descendants), a name referred to as a [Relative Name](#) is used.

ある [Functional Object](#) を起点にその [Functional Object](#) (及びその子孫) の中で定義されている物を参照するときは、[相対名](#)と称する名前を用いる。

When referring from a [Functional Object](#) to an entity defined within the [Functional Object](#), its [Relative Name](#) is its [Local Name](#).

ある [Functional Object](#) を起点にその [Functional Object](#) で定義されている物を参照するときは、[相対名](#)は、[その局所名](#)である。

When referring from a [Functional Object](#) to an entity defined in a child [Functional Object](#) of the [Functional Object](#), its [Relative Name](#) is a combination of the [Local Names](#) of the child [Functional Object](#) and the entity defined in the child [Functional Object](#) in this order connected with a period sign in between.

ある [Functional Object](#) を起点に、その [Functional Object](#) の子 [Functional Object](#) で定義されている物を参照するときは、[相対名](#)は、その子 [Functional Object](#) の局所名、子 [Functional Object](#) で定義されている物の局所名を、この順に、ピリオドを挟んで接続したものである。

Similarly, when referring from a **Functional Object** to an entity defined in a grandchild **Functional Object** of the **Functional Object**, its **Relative Name** is a combination of the **Local Names** of the child **Functional Object**, the grandchild **Functional Object**, and the entity defined in the grandchild **Functional Object** in this order connected with a period sign in between. The same rule applies to the **Relative Names** of great-grandchild **Functional Objects**, and so on.

When referring to an entity in a spacecraft, the name referred to as the **Absolute Name** is used.

The **Absolute Name** is represented by the **Relative Name** from the top-level **Functional Object** (*i.e.*, the **Functional Object** of the entire spacecraft) prefixed with a slash.

When a delivery **Route** to a destination **Functional Object** is specified for an **SMCP Telecommand** in addition to the **Operation** to be invoked, the **Absolute Name** of the **SMCP Telecommand** is a combination of the **Absolute Name** of the **Functional Object** to which the **Operation** belongs, the **Local Name** of the delivery **Route**, and the **Local Name** of the **Operation** in this order with a period sign in between.

同様に、ある **Functional Object** を起点に、その **Functional Object** の孫 **Functional Object** で定義されている物を参照するときは、**相対名**は、子 **Functional Object** の**局所名**、孫 **Functional Object** の**局所名**、及び孫 **Functional Object** で定義されている物の**局所名**を、この順に、それぞれピリオドを挟んで接続したものである。以下、ひ孫 **Functional Object** 以下の**相対名**についても同様の規則を適用する。

宇宙機内の任意の物を参照するときは、**絶対名**と称する名前を用いる。

**絶対名**は、最上位の **Functional Object** (すなわち宇宙機全体の **Functional Object**) を起点とする**局所名**にスラッシュを前置したもので表す。

ある **SMCP Telecommand** に、駆動する **Operation** に加えて送付先 **Functional Object** への配信 **Route** も指定する場合は、その **SMCP Telecommand** の**絶対名**は、その **Operation** の所属する **Functional Object** の**絶対名**、配信 **Route** の**局所名**、及び、その **Operation** の**局所名**を、この順に、ピリオドを挟んで接続したものとなる。

When referring to the value of an **Attribute** by specifying the delivery **Route** of the **SMCP Telemetries** carrying the values, the **Absolute Name** of the reference is a combination of the **Absolute Name** of the **Functional Object** that has the **Attribute**, the **Local Name** of the delivery **Route**, and the **Local Name** of the **Attribute** in this order connected with a period sign in between.

The **Parameter** name of an **Operation** or **Alert class** **shall** be unique within the **Operation** or **Alert class**, respectively.

The **State** name of a **State Machine** **shall** be unique within the **State Machine**.

The child **Functional Object** name, **Route** name, **Operation** name, **Attribute** name, **Pseudo Attribute** name, **State Machine** name, **Alert class** name, and **Attribute Sequence** name of a **Functional Object** **shall** be unique within the **Functional Object** regardless of the type of the entity referred to by the name.

The **Condition Expression/Event class** name, **Attribute Change Rule** name, **Attribute Limit** name, **Event class** name, **Engineering Unit Conversion** name, and **Valid Value Set** name of a **Functional Object** **shall** be unique within the **Functional Object** for each type of entity referred to by the name.

If an **Attribute Limit**, **Engineering Unit Conversion**, or **Valid Value Set** is used in only one **Attribute**, its **Absolute Name** **should** be identical with the **Absolute Name** of the **Attribute**; [Rationale] for easy understanding.

ある **Attribute** の値を、その値を運ぶ **SMCP Telemetries** の配送 **Route** を指定して参照する場合、その参照パターンの絶対名は、その **Attribute** を持つ **Functional Object** の絶対名、配信 **Route** の局所名、及び、**Attribute** の局所名を、この順に、それぞれピリオドを挟んで接続したものとなる。

ある **Operation** または **Alert class** の **Parameter** 名は、それぞれ、その **Operation** または **Alert class** においてユニークであること。

ある **State Machine** の **State** 名は、その **State Machine** においてユニークであること。

ある **Functional Object** の子 **Functional Object** 名、**Route** 名、**Operation** 名、**Attribute** 名、**Pseudo Attribute** 名、**State Machine** 名、**Alert class** 名、**Attribute Sequence** 名は、その **Functional Object** において、名前で参照する物の種類によらずユニークであること。

ある **Functional Object** の名前付条件式・**Event class** 名、**Attribute Change Rule** 名、**Attribute Limit** 名、**Event class** 名、**工学値変換**名、**Valid Value Set** 名は、その **Functional Object** において、名前で参照する物の種類毎にユニークであること。

ある **Attribute Limit**, **工学値変換**、または **Valid Value Set** が、一つの **Attribute** のみで用いられる場合、これらの絶対名は、その **Attribute** の絶対名と同一であるべきである；[根拠] 分かりやすくするため。

If an **Attribute Change Rule** is used for only one **Operation**, its **Absolute Name** **should** be identical with the **Absolute Name** of the **Operation**; [Rationale] for easy understanding.

ある **Attribute Change Rule** が、一つの **Operation** のみで用いられる場合、その**絶対名**はその **Operation** の**絶対名**と同一であるべきである; [根拠] 分かりやすくするため。

### 5.3. Rules for assigning IDs // IDs 割り当て規則

The ID values used in **SMCP Telecommands** and **SMCP Telemetries** **shall** be unique within each **Functional Object** for each type of ID.

**SMCP Telecommands** 及び **SMCP Telemetries** 内で用いる ID の値は、それぞれの ID の種類毎に各 **Functional Object** 内でユニークであること。

[Note] The **Attribute IDs** used for **Attributes** (see Section 3.2.3.2) and for **Attribute Sequences** (see Section 3.7.2) share the common ID space.

[注] **Attributes** (3.2.3.2 項参照) と **Attribute Sequences** (3.7.2 項参照) に用いる **Attribute IDs** は、共通の ID 空間を共有する。

The encoded values of the **States** in **State Machines** **shall** be unique within each **State Machine**.

**State Machines** の **State** のエンコード値は、各 **State Machine** 内でユニークであること。

The values of **Valid Value Sets** **shall** be unique within each **Valid Value Set**.

**Valid Value Sets** の値は、各 **Valid Value Set** 内でユニークであること。

#### 5.4. Telemetry Item Name, Telecommand Name, and Command Name Telemetry 項目名、Telecommand 名、及び Command 名

The naming rules of Telemetry Item Names, Telemetry 項目名、Telecommand Names、及び Command Names Telecommand 名、及び Command 名の命名規則を Table 5-1 に列挙する。なお、絶対名による表記を基本とするが必要に応じ相対名を用いて表記する事もある。

are listed in Table 5-1. The Absolute Names are used basically, but Relative Names might be used if necessary.

**Table 5-1: Naming Rules of Telemetry Item Names, Telecommand Names, and Command Names // Telemetry 項目名、Telecommand 名、及び Command 名の命名規則**

Name 名称	Relative Name 局所名	Example of Absolute Name 絶対名の例
		<p>Functional Object name : /A.B</p> <p>Route name : C</p> <p>Functional Object 名 : /A.B</p> <p>Route 名 : C</p>
Telemetry Item Name Telemetry 項目名	<p>Relative Name of Attribute Attribute の局所名</p>	<p>Attribute name : TLM</p> <p>Absolute Name (without Route specification) : /A.B.TLM</p> <p>Attribute 名 : TLM</p> <p>絶対名 (Route 指定無) : /A.B.TLM</p>
ACTION Telecommand name ACTION Telecommand 名	<p>Relative Name of Operation Operation の局所名</p>	<p>Operation name : CMD</p> <p>Absolute Name (with Route specification) : /A.B.C.CMD</p> <p>Operation 名 : CMD</p> <p>絶対名 (Route 指定有) : /A.B.C.CMD</p>
SET Telecommand name SET Telecommand 名	<p>SET_ + Attribute (Sequence) name SET_ + Attribute (Sequence) 名</p>	<p>Attribute (Sequence) name : TLM</p> <p>Absolute Name (without Route specification) : /A.B.SET_TLM</p> <p>Attribute (Sequence) 名 : TLM</p> <p>絶対名 (Route 指定無) : /A.B.SET_TLM</p>
GET Telecommand name GET Telecommand 名	<p>GET_ + Attribute (Sequence) name GET_ + Attribute (Sequence) 名</p>	<p>Attribute (Sequence) name : TLM</p> <p>Absolute Name (with Route specification) : /A.B.C.GET_TLM</p> <p>Attribute (Sequence) 名 : TLM</p> <p>絶対名 (Route 指定有) : /A.B.C.GET_TLM</p>

		<b>Memory Functional Object name : /A.B</b> <b>Route name : C</b> <b>Memory Functional Object 名 : /A.B</b> <b>Route 名 : C</b>
<b>WRITE Command name</b>  <b>WRITE Command 名</b>	<b>WRITE</b>	<b>Absolute Name (without Route specification) : /A.B.WRITE</b> <b>絶対名 (Route 指定無) : /A.B.WRITE</b>
<b>LOAD Command name</b>  <b>LOAD Command 名</b>	<b>LOAD</b>	<b>Absolute Name (without Route specification) : /A.B.LOAD</b> <b>絶対名 (Route 指定無) : /A.B.LOAD</b>
<b>READ Command name</b>  <b>READ Command 名</b>	<b>READ</b>	<b>Absolute Name (with Route specification) : /A.B.C.READ</b> <b>絶対名 (Route 指定有) : /A.B.C.READ</b>
<b>DUMP Command name</b>  <b>DUMP Command 名</b>	<b>DUMP</b>	<b>Absolute Name (with Route specification) : /A.B.C.DUMP</b> <b>絶対名 (Route 指定有) : /A.B.C.DUMP</b>

## 6. Diagnostic Rule Definition // Diagnostic Rule 定義

### 6.1. General // 一般

The Diagnostic Rule Definition is the definition to specify Diagnostic Rules. Diagnostic Rule 定義は、Diagnostic Rules を定めるための定義である。

### 6.2. Diagnostic Rule

For a Diagnostic Rule, the following items **shall** be defined:

- 1) Diagnostic Rule name  
 ::= the name of the Diagnostic Rule
- 2) Condition Expression  
 See Section 3.1.4
- 3) Referenced Attribute name (0~n)
- 4) Message to human operator (text)

[Note] It is possible to define much more complex diagnostic rules than those described above by using Automated Monitoring Software (ATMOS).

各 Diagnostic Rule に、以下の項目を定義すること。

- 1) Diagnostic Rule name  
 ::= Diagnostic Rule の名前
- 3) 条件式  
 3.1.4 項参照
- 4) 参照 Attribute name (0~n)
- 5) 運用者へのメッセージ (テキスト)

[注] 自動監視ソフトウェア (Automated Monitoring Software, ATMOS) を用いる事で、上記に記載したものよりもはるかに複雑な診断ルールを定義する事が可能である。

## 7. Other Subsidiary Definitions // その他の副次的な定義

### 7.1. Spacecraft Information Definition // 宇宙機情報定義

The **Spacecraft Information Definition** defines information that applies to all **Functional Objects** aboard a spacecraft.

For a spacecraft, the following item **shall** be defined.

1) **Upper APID Type**

$::=$  the type of assignment of the **Upper APID** (Type 1a or Type 1b; see Section 5.2.3 in [A4])

**宇宙機情報定義 (Spacecraft Information Definition)** は、宇宙機の全ての **Functional Objects** に適用する情報を定義する。

各宇宙機に、以下の項目を定義すること。

1) **Upper APID Type**

$::=$  **Upper APID** の割り当てタイプ (Type 1a または Type 1b; [A4] 5.2.3 項参照)

## 7.2. MAP ID Definition // MAP ID 定義

### 7.2.1. General // 一般

The **MAP ID Definition** is the definition to specify the **MAP ID** of the telecommand.

**MAP ID 定義 (MAP ID Definition)** は、テレコマンドの **MAP ID** を定める定義である。

### 7.2.2. MAP ID

For a spacecraft, the following items **shall** be defined:

- 1) **Map Channel** presence/absence  
 ::= whether the **MAP Channel** is used for telecommand transmission or not
- 2) Default **MAP ID** (optional)  
 ::= default **MAP ID**  
  
If the **Map Channel** is used, this item **shall** be specified. Otherwise, this item **shall** not be specified.

各宇宙機に対して、以下の項目を定義すること。

- 1) **Map Channel** 有無  
 ::= テレコマンドの伝送に **MAP Channel** を使用するか否か
- 2) デフォルト **MAP ID** (optional)  
 ::= デフォルトの **MAP ID**  
  
**Map Channel** を使用する場合、本項目を指定すること。さもなければ、本項目は指定しないこと。

#### 7.2.2.1. Telecommand with MAP ID specification // MAP ID 指定有りの Telecommand

For a telecommand that uses a **MAP ID** other than the default one, the following items **shall** be defined:

- 1) **Telecommand Name**  
 ::= **Absolute Name** of a telecommand that uses a **MAP ID** other than the default one
- 2) **MAP ID**  
 ::= **MAP ID** value

デフォルト以外の **MAP ID** を使用する各テレコマンドに、以下を定義すること。

- 1) **Telecommand 名**  
 ::= デフォルト以外の **MAP ID** を使用するテレコマンドの**絶対名**
- 2) **MAP ID**  
 ::= **MAP ID** の値

### 7.3. Telemetry Packet Length Information Definition // テレメトリパケット長情報定義

The Telemetry Packet Length Information テレメトリパケット長情報は、テレメトリのパDefinition is the definition to specify the ケット長を定める定義である。  
lengths of telemetry packets.

For an APID, the following item **shall** be defined.

- |  |   |
|--|---|
| 1) <u>Maximum Packet Length</u><br>::= the maximum length of a Space Packet<br>with the APID | 1) <u>最大パケット長</u><br>::= 同 APID を持つ Space Packet の最大長 |
|--|---|

## 8. APPENDIX A. ACRONYMS // 略語

This chapter lists the acronyms used in this document.

本章では、本書が用いる略語一覧を示す。

ACK	Acknowledge
ACT	Action
APID	Application Process Identifier
AS	Attribute Sequence
ATMOS	Automated Monitoring Software
CISW	Telecommand Issue Software
CLCW	Communications Link Control Word
COP	Communications Operation Procedure
FMS	Functional Model of Spacecrafts
FOID	Functional Object Identifier
GSTOS	Generic Spacecraft Test and Operations Software
ID	Identifier
MAP	Multiplexer Access Point
NTF	Notification
SCDHA	Standard of Communications and Data-Handling Architecture
SIB2	Spacecraft Information Base 2
SMCP	Spacecraft Monitor and Control Protocol
SW	Software
TC	Telecommand
UFORID	Upper Functional Object and Route Identifier
XML	Extensible Markup Language