

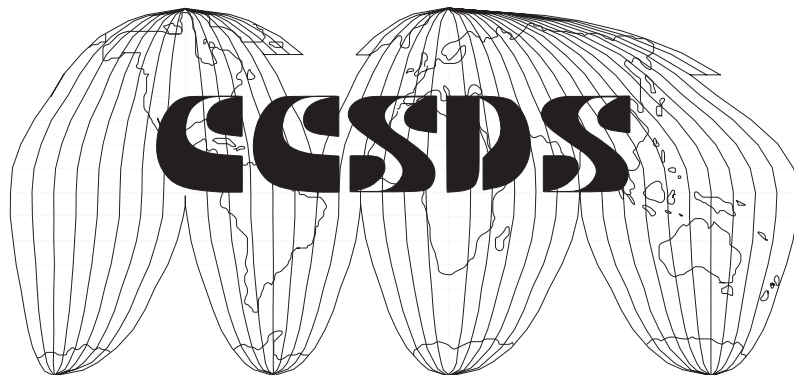
***Consultative
Committee for
Space Data Systems***

**RECOMMENDATION FOR SPACE
DATA SYSTEM STANDARDS**

**DATA ENTITY DICTIONARY
SPECIFICATION LANGUAGE (DEDSL)—
ABSTRACT SYNTAX
(CCSD0011)**

**CCSDS 647.1-B-1
BLUE BOOK**

June 2001



AUTHORITY

Issue:	Blue Book, Issue 1
Date:	June 2001
Location:	Oxfordshire, England

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS Recommendations is detailed in [C1], and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Program Integration Division (Code MT)
National Aeronautics and Space Administration
Washington, DC 20546, USA

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organisation officially established by the management of member space Agencies. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommendations** and are not considered binding on any Agency.

This **Recommendation** is issued by, and represents the consensus of, the CCSDS Plenary body. Agency endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- o Whenever an Agency establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommendation**. Establishing such a **standard** does not preclude other provisions which an Agency may develop.
- o Whenever an Agency establishes a CCSDS-related **standard**, the Agency will provide other CCSDS member Agencies with the following information:
 - The **standard** itself.
 - The anticipated date of initial operational capability.
 - The anticipated duration of operational service.
- o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommendation** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommendation** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or, (3) be retired or canceled.

In those instances when a new version of a **Recommendation** is issued, existing CCSDS-related Agency standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each Agency to determine when such standards or implementations are to be modified. Each Agency is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommendation.

FOREWORD

This Recommendation is a technical Recommendation that provides a model and language to increase the standardisation of the expression of semantic concepts that are to be carried with data. These semantic concepts are given standard names, and a standard way of expressing them is also provided. The semantic information may be conveyed either in a computer-processable manner or via conventional (e.g., paper) documentation.

Through the process of normal evolution, it is expected that expansion, deletion or modification to this document may occur. This Recommendation is therefore subject to CCSDS document management and change control procedures defined in reference [C1]. Current versions of CCSDS documents are maintained at the CCSDS Web site:

<http://www.ccsds.org/>

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

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

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centre National d’Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA HQ)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Industry Canada/Communications Research Centre (CRC)/Canada.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title and Issue	Date	Status
CCSDS 647.1-B-1.	Data Entity Dictionary Specification Language (DEDSL)—Abstract Syntax	June 2001	Original Issue

CONTENTS

<u>Section</u>	<u>Page</u>
1 INTRODUCTION.....	1-1
1.1 PURPOSE AND SCOPE.....	1-1
1.2 APPLICABILITY.....	1-2
1.3 RATIONALE.....	1-2
1.4 DOCUMENT STRUCTURE.....	1-3
1.5 DEFINITIONS.....	1-4
1.6 REFERENCES.....	1-7
2 OVERVIEW	2-1
2.1 GENERAL.....	2-1
2.2 USES OF DATA ENTITY DICTIONARIES.....	2-1
2.3 APPLICATION OF THE DEDSL.....	2-6
2.4 REGISTERING DED	2-8
3 DESCRIPTORS OF A DATA ENTITY ATTRIBUTE.....	3-1
3.1 GENERAL.....	3-1
3.2 LIST OF DESCRIPTORS	3-1
3.3 ATTRIBUTE_NAME.....	3-3
3.4 ATTRIBUTE_DEFINITION.....	3-4
3.5 ATTRIBUTE_OBLIGATION.....	3-5
3.6 ATTRIBUTE_CONDITION.....	3-6
3.7 ATTRIBUTE_MAXIMUM_OCCURRENCE.....	3-6
3.8 ATTRIBUTE_VALUE_TYPE.....	3-7
3.9 ATTRIBUTE_MAXIMUM_SIZE	3-9
3.10 ATTRIBUTE_ENUMERATION_VALUES	3-10
3.11 ATTRIBUTE_COMMENT.....	3-10
3.12 ATTRIBUTE_INHERITANCE.....	3-11
3.13 ATTRIBUTE_DEFAULT_VALUE.....	3-11
3.14 ATTRIBUTE_VALUE_EXAMPLE.....	3-12
3.15 ATTRIBUTE_SCOPE.....	3-12
4 DATA ENTITY ATTRIBUTES AND DATA ENTITY DICTIONARY ATTRIBUTES.....	4-1

4.1	CONCEPT DEFINITIONS.....	4-1
4.2	GENERAL VIEW OF THE STANDARD ATTRIBUTES	4-2
4.3	STANDARD ATTRIBUTES FOR DATA ENTITY DICTIONARIES	4-4
4.4	STANDARD ATTRIBUTES FOR DATA ENTITIES	4-13
4.5	USER-DEFINED ATTRIBUTES.....	4-34
4.6	RELATIONSHIP RULES	4-37
5	IMPLEMENTATION GUIDELINES	5-1
6	DEDSL CONFORMANCE: ABSTRACT DEDSL (ADID = CCSD0011).....	6-1
6.1	GENERAL.....	6-1
6.2	CONFORMANCE LEVEL 1: BASE COMPLIANCE	6-1
6.3	CONFORMANCE LEVEL 2: FULL COMPLIANCE	6-1
ANNEX A	DEDSL EXAMPLES	A-1
ANNEX B	MAPPING OF THE CONCEPTS BETWEEN THIS RECOM- MENDATION AND THE ISO/IEC 11179-3:1994 STANDARD	B-1
ANNEX C	INFORMATIVE REFERENCES.....	C-1

Figure

2-1	Organisation of the Data Product Product_X.....	2-2
2-2	Organisation of the Data Product Product_Y.....	2-3
2-3	Organisation of the DED Relative to Product_X and Product_Y.....	2-4
2-4	“Uses some models of” Links between Data Product Dictionaries	2-6

Table

2-1	Comparison of Community DED and Product DED Descriptions.....	2-5
3-1	General Descriptors.....	3-2
4-1	Data Entity Dictionary Attributes	4-2
4-2	Data Entity Attributes	4-3

Example

3-1	Elementary Type	3-7
3-2	Ordered List of Elementary Types.....	3-8
3-3	Ordered List of the Same Elementary Type.....	3-8
3-4	Use of Entity_Type	3-8
3-5	Use of Choice.....	3-8

CONTENTS (continued)

<u>Example</u>	<u>Page</u>
3-6 Use of Attribute_Maximum_Size	3-9
3-7 Use of Attribute_Enumeration_Values	3-10
4-1 Defining User-Defined Attribute	4-35
4-2 Defining User-Defined Attribute with an External Reference.....	4-35
4-3 Usage of a User-Defined Attribute	4-36

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

This document provides an extensible way of defining data entity dictionaries.

A Data Entity is a concept that can, or does, take on one or more values. Semantics of a data entity, such as a text definition of its meaning, are defined by attributes. The purpose of this Recommendation is to define a language for specifying a dictionary which describes semantics for a collection of data entities—it does not define a specific dictionary.

A dictionary is understood as a mechanism that is able to organise a set of information in a consistent and easily understandable manner, and it is commonly used by humans to look up the meaning of words used in natural languages. Similarly, a Data Entity Dictionary (DED) is used by humans and systems to look up the definition, and other attributes, of data entities used in the definition and generation of data products.

This document defines the abstract definition of the semantic information that is required to be conveyed and presents the specification in a layered manner (attributes, entities, dictionaries). This is done so that the actual technique used to convey the information is independent of the information content and, therefore, the same abstract standard can be used within different formatting environments. This also permits the semantic information to be translated to different representations as may be needed when data are transferred across different domains.

This Recommendation defines the concepts of name, definition, units, and a small set of other standard attributes so that they may be used consistently in the formation of data entity dictionaries. Given the wide variety of data entities that may need to be described, only a few of the attributes are made mandatory by this Recommendation.

The method used to define standard attributes can also be used to extend the set of attributes beyond the standard ones provided within this Recommendation.

Several classes of data entities are defined. These classes allow making a distinction between the abstract data entities—the models—and the concrete data entities—the data fields in a data product.

This Recommendation is strongly based on the International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) Specification and standardisation of data elements (reference [C11]) with which it is widely consistent for all semantic aspects. **However this Recommendation does not require knowledge of reference [C-11] in order to be understood.**

1.2 APPLICABILITY

This Recommendation is intended to be used by implementers of a concrete syntax for Data Entity Dictionaries. Such dictionaries could then be used for example:

- by data producers to construct dictionaries that describe, in a more formal manner, data entities within their data products;
- by data users to understand data received from data producers who have used this Recommendation to construct their dictionaries;
- by an organisation that mandates the attributes used to define each entity description in dictionaries used within that organisation;
- by a particular community, such as Earth observation, space physics, archives, etc., to establish a degree of standardisation for the contents of any data entity dictionary associated or not with a data product (this would be done by using this Recommendation to define a community-wide data entity dictionary);
- by organisations and communities to exchange the contents of a data entity dictionary in a standardised manner, i.e., to facilitate interoperability.

1.3 RATIONALE

A given data entity may take on a range of values that are represented differently within different formats, including such generic formats Hierarchical Data Format (HDF) (reference [C-6]), Common Data Format (CDF) (reference [C-7]), or in native formats. However there is information about that data entity, such as its definition and other semantic attributes, which is independent of the values and their representation in any given format.

Part of this information can be expressed in a data entity dictionary, and this dictionary can be expressed in many different ways. For example, it may be expressed in natural language paragraphs within a document that accompanies a data product. It may be partially expressed by attributes defined within generic or native data product formats, while the rest may be in other documents. The concepts used in the description of the data entities may vary widely, or subtly. These concepts may not be documented at all.

Therefore, individuals and organisations that need to receive and understand a variety of data products may waste considerable effort in attempting to understand the data entities comprised by each data product. This also greatly hinders the use of generic tools that can assist in the recognition and presentation of this information in a way that various individuals and organisations find most understandable.

To begin to address these issues across broad organisation and community disciplines, it is first necessary to define a set of standard concepts that can be used in the formation of data entity dictionaries within those disciplines, and in the mapping of different dictionary concepts between disciplines. To facilitate the creation of generic tools, it is also necessary to define a standard representation for the standard concepts. This Recommendation defines a small number of concepts, in terms of attribute descriptors and standard attributes, which are intended to be broadly applicable.

Finally, this Recommendation provides additional standardised functionalities allowing the expression of relationships between entities in a dictionary, as well as definitions of inter-relationships between dictionaries.

1.4 DOCUMENT STRUCTURE

The document is structured as follows:

- Section 2 provides an overview of the data entity dictionary concept and describes through examples how this Recommendation may be used. It also provides a context enabling a better understanding of the standard specified in sections 3 and 4.
- Section 3 specifies the standard descriptors which shall be used to define the attributes of data entities as well as of data entity dictionaries.
- Section 4 specifies the standard attributes to be used to define data entities and data entity dictionaries. It also presents the method to define user-defined attributes and the implementation guidelines.
- Section 5 discusses implementation guidelines.
- Section 6 discusses levels of conformance with this Recommendation.
- Annex A provides examples implementing the concepts described in sections 3 and 4.
- Annex B provides the mapping of the concepts between this Recommendation and the ISO/IEC 11179-3:1994 standard (reference [C-11]).
- Annex C provides a list of references that may be valuable to the user of this Recommendation as background material or as implementation guidelines for using this Recommendation.

1.5 DEFINITIONS

1.5.1 ACRONYMS AND ABBREVIATIONS

This subsection defines the acronyms and abbreviations used throughout this Recommendation:

ADID	Authority and Description Identifier
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
CDF	Common Data Format
DDL	Data Definition Language
DED	Data Entity Dictionary
DEDSL	Data Entity Dictionary Specification Language
EAST	Enhanced Ada SubseT
FITS	Flexible Image Transport System
HDF	Hierarchical Data Format
ID	Identifier
ISO	International Organization for Standardization
LVO	Label Value Object
MACAO	Member Agency Control Authority Office
PVL	Parameter Value Language
SFDU	Standard Formatted Data Unit
XML	eXtended Mark-up Language

1.5.2 GLOSSARY OF TERMS

For the purpose of this document, the following definitions apply:

Attribute	A piece of information that describes a Data Entity or Dictionary Entity. This information characterizes or enhances the understanding of the data that is described. Attributes are used to define the semantics of data entities.
Attribute Descriptor	A piece of information that describes an attribute. This document specifies a set of descriptors for attribute description.
Attribute Value	A value associated with an attribute instance.
Composite Data Entity	A data entity which consists of a combination of various other elementary and composite entities.
Constant	A named constant value that is used within a dictionary but is not part of the data themselves. Use of constants enables data entity dictionaries to specify values which will be used by several projects or within a domain (astronomy constants, image size, etc).

Data Entity	A concept that can, or does, take on one or more values. The concept, and optionally constraints on the representation of its value, are defined by attributes and their values.
Data Entity Dictionary	A collection of semantic definitions of various data entities, together with a few mandatory and optional attributes about the collection as a whole. Data entity dictionaries may be just for a single product, i.e., all the data entities within a single product are described in a corresponding single dictionary, or the data entity dictionary may be a discipline-oriented dictionary that holds a number of previously defined data entity definitions which may be used by data designers and users as references. Some parts of a dictionary are optional. In practical terms the dictionary could be a file or a Standard Formatted Data Unit (SFDU) Label-Value Object (LVO) value field (references [1] and [C-4]). Within this Recommendation, the expression ‘data entity dictionary’ can refer either to the notion of data entity dictionaries or to a data entity dictionary instance. A data entity dictionary is also an entity, called Dictionary Entity.
Data Product	A collection of one or more data items that are packaged for or by a specific application.
Defaulted	Indication of an attribute or descriptor value that is understood when the attribute or descriptor is not explicitly included in the containing definition.
Descriptor Name	An Identifier that is the name of the descriptor.
Descriptor Type	The characterization of the descriptor value; e.g., text, identifier, integer.
Elementary Data Entity	A data entity whose data type is elementary, that is, Integer, Real, Text or Enumerated.
Enumerated	A set containing a restricted number of discrete values, where each discrete value is named and unique within the set.
Identifier	A sequence of characters that designates something.
Inter-operability Constraints	Set of constraints enabling an easier exchange of dictionaries using different DEDSL implementations.
Model	A data entity described independently from any instance in a data product, and corresponding to a re-usable data entity definition, from which other data entities may inherit the attributes and apply some specialization rules.

Semantics	Information that defines the meaning rather than the physical representation of data. Semantics potentially cover a very large domain, from the simple domain, such as the units of one data entity, to a more complex one, such as the relationship between one data entity and another.
Standard Attribute	One of the attributes defined within this Recommendation.
Syntax	Information defining the physical representation of data. It includes the structural arrangement of the fields within the data on the exchanged media.
Text	A delimited sequence of characters. The set of allowed characters is defined in the Data Entity Dictionary.
User Defined Attribute	An attribute that is defined by a particular user or project and after definition is then used in the same manner as a Standard Attribute within that data entity dictionary.

1.5.3 NOMENCLATURE

The following conventions apply throughout this Recommendation:’

- a) the words ‘shall’ and ‘must’ imply binding and verifiable specification;
- b) the word ‘should’ implies an optional, but desirable, specification;
- c) the word ‘may’ implies an optional specification;
- d) the words ‘is’, ‘are’ and ‘will’ imply statements of fact.

1.5.4 CONVENTIONS

Convention 1:

No **Identifiers** or encoding values in the document are case-sensitive.

Convention 2:

The words which appear in bold characters throughout this Recommendation correspond to the keywords or the major concepts of this Recommendation.

For example, when defining the descriptors, the descriptor name as well as the descriptor type and sometimes the predefined values of the type are in bold characters to enlighten the definition. This convention also applies when defining the attributes (respectively attribute name and attribute value type).

Convention 3:

Text values are delimited by quotes in the examples for clarity.

Convention 4:

Parentheses are used to delimit ordered lists, with commas used to separate the elements of the list.

Convention 5:

Interoperability constraints are a set of constraints enabling an easier exchange of dictionaries using different DEDSL implementations and may appear within the description of the descriptors and attributes specified by this Recommendation. For example, they may refer to the size of the value of an attribute or the coding values of a descriptor. Interoperability constraints determine the two levels of conformance identified in section 6. Interoperability constraints are introduced by the title '*Interoperability Constraints*'.

1.6 REFERENCES

The following documents contain provisions (through references within this text) which constitute provisions of this Recommendation. At the time of the publication the indicated editions were valid. All documents are subject to revision, and users of this Recommendation are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently available CCSDS Recommendations.

- [1] *Standard Formatted Data Units—Structure and Construction Rules*. Recommendation for Space Data System Standards, CCSDS 620.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, May 1992. (ISO 12175)
- [2] *ASCII Encoded English (CCSD0002)*. Recommendation for Space Data System Standards, CCSDS 643.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, November 1992. (ISO 14962)
- [3] *Information Processing—Representation of numerical values in character strings for information interchange*. ISO 6093-1985. Geneva: ISO, 1985.
- [4] *Standard Formatted Data Units—Control Authority Procedures*. Recommendation for Space Data System Standards, CCSDS 630.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, June 1993. (ISO 13764)
- [5] *Codes for the Representation of Names of Languages*. ISO 639-2 -1998. Geneva: ISO, 1998.
- [6] *Information Processing—8-Bit Single-Byte Coded Graphic Character Sets—Part 1: Latin Alphabet No. 1*. International Standard, ISO 8859-1:1987. Geneva: ISO, 1987.
- [7] *Information Processing—Representation of SI and Other Units in Systems with Limited Character Sets*. International Standard, ISO 2955-1983. Geneva: ISO, 1983.

2 OVERVIEW

2.1 GENERAL

As discussed in subsection 1.1, a Data Entity Dictionary is used by humans and systems to look up the definition and other attributes of data entities. This section discusses some of the primary uses of data entity dictionaries and presents a basic example of DEDSL usage.

Other detailed examples are given in annex A.

2.2 USES OF DATA ENTITY DICTIONARIES

2.2.1 PRODUCT DATA ENTITY DICTIONARIES

So that its contained data can be extracted, a data product can appear with a formatting standard (e.g., Flexible Image Transport System [FITS]), or a self-describing format (e.g., CDF, HDF, etc.) or a Data Definition Language (e.g., EAST). This syntactic information may not be easy to understand and a formal definition of additional semantics may be necessary, which leads to the definition of a product Data Entity Dictionary.

Figure 2-1 shows the structure of the data product (**PRODUCT_X**). It is made up of a header (**HEADER**) and an image (**DATA_1**). The header contains a product identifier (**PRODUCT_ID**), information about the station, which has acquired the data, (**ACQ_STATION**), information about the acquisition time (**ACQ_TIME**), and information necessary for the processing of the image, e.g., the centre coordinates [**CENTRE_COORD** (**LATITUDE** and **LONGITUDE**)]. Its physical description, possibly expressed using a DDL, may not be readily understandable to all readers because it includes a specification to the bit level. It is also useful to have a quick overview of the product and to have additional semantic information. Therefore, the definition of a product data entity dictionary (**PRODUCT_X DED**) will bring this necessary information and perspective.

Within this data entity dictionary additional information can be given to more precisely define:

- the definition of each data entity;
- the kinds of values used for the centre coordinates **LATITUDE** and **LONGITUDE**.

The values of **LATITUDE** are defined relative to the Equator and range from -90.000 to +90.000, while the values of **LONGITUDE** are defined to be relative to Greenwich and range from -180.00 to +180.00. The units used to express the values are to be in degrees. All this information may be included in the data entity dictionary.

These pieces of semantic information correspond to the existing data entities of the product.

Consequently, the **PRODUCT_X DED** will contain the semantic descriptions of the following data entities:

- HEADER;
- PRODUCT_ID;
- ACQ_STATION;
- ACQ_TIME;
- CENTRE_COORD;
- LATITUDE;
- LONGITUDE;
- DATA_1.

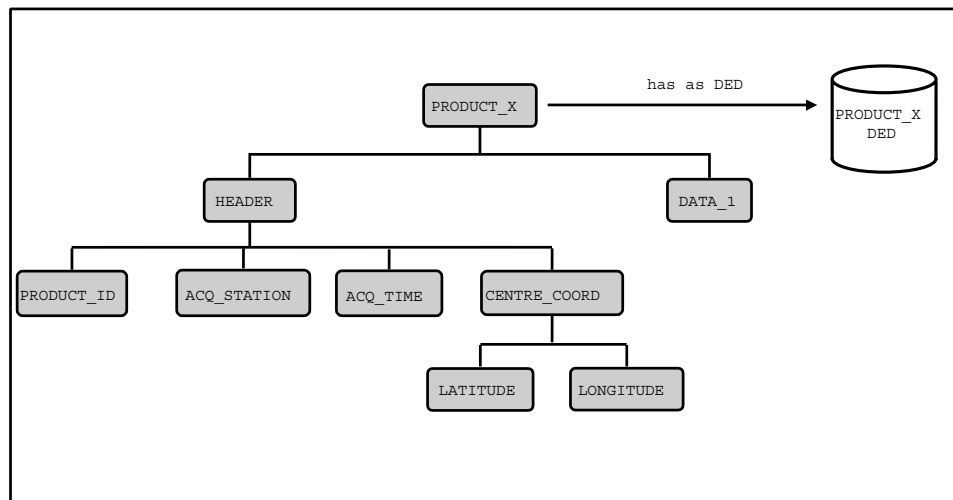


Figure 2-1: Organisation of the Data Product Product_X

Figure 2-2 shows the structure of another data product (**PRODUCT_Y**). It is made up of the product identifier (**PRODUCT_ID**), the centre coordinates (e.g., a latitude, a longitude) and the acquired image (**DATA_2**). A data product dictionary (**PRODUCT_Y DED**) can also be defined for **PRODUCT_Y** in the same way as for **PRODUCT_X**.

Consequently, the **PRODUCT_Y DED** will contain the semantic descriptions of the following data entities:

- PRODUCT_ID;
- LATITUDE;
- LONGITUDE;
- DATA_2.

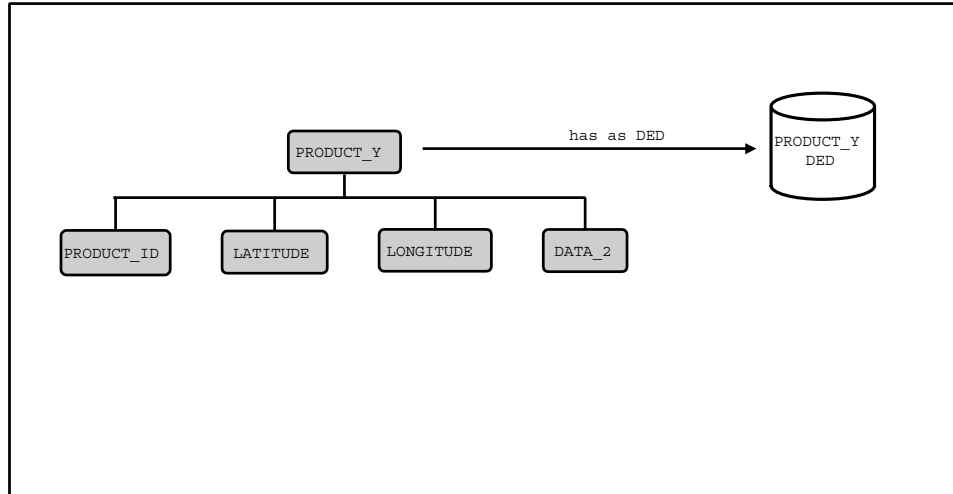


Figure 2-2: Organisation of the Data Product Product_Y

Looking at the two previously described data products it seems convenient to define **PRODUCT_Y DED** by re-using some data entity descriptions of **PRODUCT_X DED**.

The data entity descriptions which seem common to both data products are **PRODUCT_ID**, **LATITUDE** and **LONGITUDE**.

Therefore the data product dictionary **PRODUCT_X** can be modified so that those data entity descriptions become re-usable **models**. These models are abstract data descriptions to which concrete descriptions, i.e., corresponding to data entities within the data product, can refer. Then the DED associated with **PRODUCT_Y** can refer to the DED associated with **PRODUCT_X** for the definition of some of its semantic descriptions using the models of the **PRODUCT_X DED**.

Consequently, the **PRODUCT_X DED** will contain data entity descriptions corresponding to abstract definitions (models) and named as follows:

- **PRODUCT_ID_MODEL**;
- **LATITUDE_MODEL**;
- **LONGITUDE_MODEL**.

The **PRODUCT_X DED** will still contain the semantic descriptions corresponding to the following data entities, but with references to the newly defined models:

- **HEADER**;
- **PRODUCT_ID**, inheriting from the model **PRODUCT_ID_MODEL**;
- **ACQ_STATION**;
- **ACQ_TIME**;

- CENTRE_COORD;
- LATITUDE, inheriting from the model LATITUDE_MODEL;
- LONGITUDE, inheriting from the model LONGITUDE_MODEL;
- DATA_1.

The **PRODUCT_Y DED** will still contain the semantic descriptions corresponding to the following data entities, but with references to the newly defined models:

- PRODUCT_ID, inheriting from the model PRODUCT_ID_MODEL;
- LATITUDE, inheriting from the model LATITUDE_MODEL;
- LONGITUDE, inheriting from the model LONGITUDE_MODEL;
- DATA_2.

Figure 2-3 presents the resulting organisation between both DEDs as they are used to support **PRODUCT_Y**.

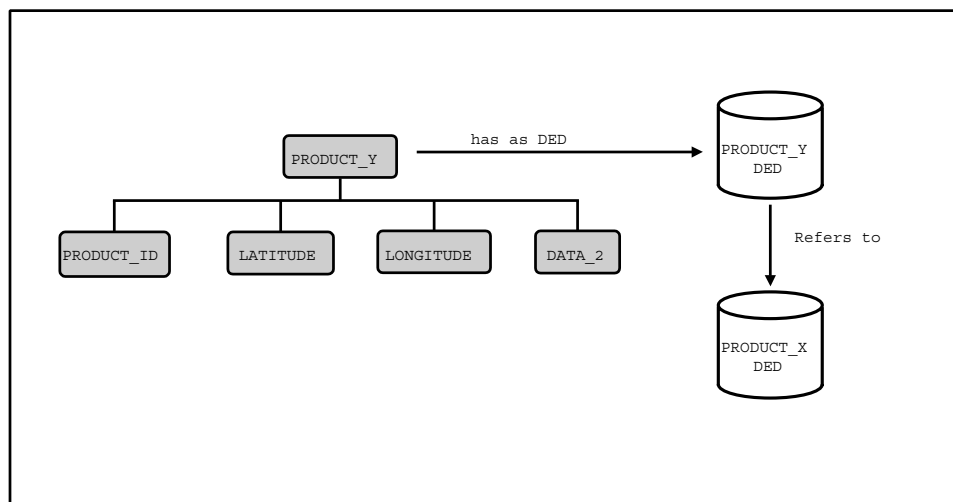


Figure 2-3: Organisation of the DED Relative to Product_X and Product_Y

2.2.2 COMMUNITY DATA ENTITY DICTIONARIES

The project or the data designer may consider that a community dictionary is necessary because there are several data products related to the same kind of data.

Looking at the two products given as examples, they may decide that the community data entity dictionary should include the following entities which frequently appear: **PRODUCT_ID**, **ACQ_STATION**, **ACQ_TIME**, **LATITUDE** and **LONGITUDE**.

This community DED will contain a normalized description of these entities which can then be considered as **models**. Data entities being latitudes and longitudes and appearing within other data products will then have the same associated semantic information whenever they inherit from these normalized descriptions.

The other data entities only appearing in a data product such as **DATA_1**, **DATA_2** and **CENTRE_COORD** are local definitions.

The purpose of a community dictionary is to provide, across different data products, a standard or normalized definition of data entities.

Table 2-1 shows an example of mapping of the community DED entries into data product DED entries for the data products shown in figures 2-1 and 2-2, according to the choices made by the project or data designers.

Table 2-1: Comparison of Community DED and Product DED Descriptions

Concept	Data Type	Found in Community DED	Found in PRODUCT_X DED	Found in PRODUCT_Y DED
HEADER	Composite	no	yes	no
PRODUCT_ID	Text	yes	yes	yes
ACQ_STATION	Enumeration	yes	yes	no
ACQ_TIME	Composite	yes	yes	no
CENTRE_COORD	Composite	no	yes	no
LATITUDE	Real	yes	yes	yes
LONGITUDE	Real	yes	yes	yes
DATA 1	Composite (Array of 16-bit integers)	no	yes	no
DATA 2	Composite (Array of real numbers)	no	no	yes

Supposing that the project has defined its community dictionary, when it defines a new data product or for example when it rewrites the dictionary related to **PRODUCT_X** (in figure 2-1), it can decide to define the **HEADER** entity on the basis of **PRODUCT_ID**, **ACQ_STATION** and **ACQ_TIME**, which inherit from the corresponding data entities in the community dictionary,

i.e., which then have the same properties as the model data entities. The project can also decide to define the **CENTRE_COORD** entity on the basis of the **LATITUDE** and **LONGITUDE** entities, which inherit from the corresponding data entities in the community dictionary. The **DATA1** entity does not inherit from a specific data entity described in the community DED, as it appears to be a kind of data only appearing in the data product **PRODUCT_X**.

The same policy can be applied to **PRODUCT_Y** of figure 2-2.

Therefore we can consider the links ‘uses some models of’, in figure 2-4, among the three data entity dictionaries. A ‘uses some models of’ link between **PRODUCT_X DED** and the domain DED means that some data entities of **PRODUCT_X** inherit from corresponding data entities contained in the domain DED.

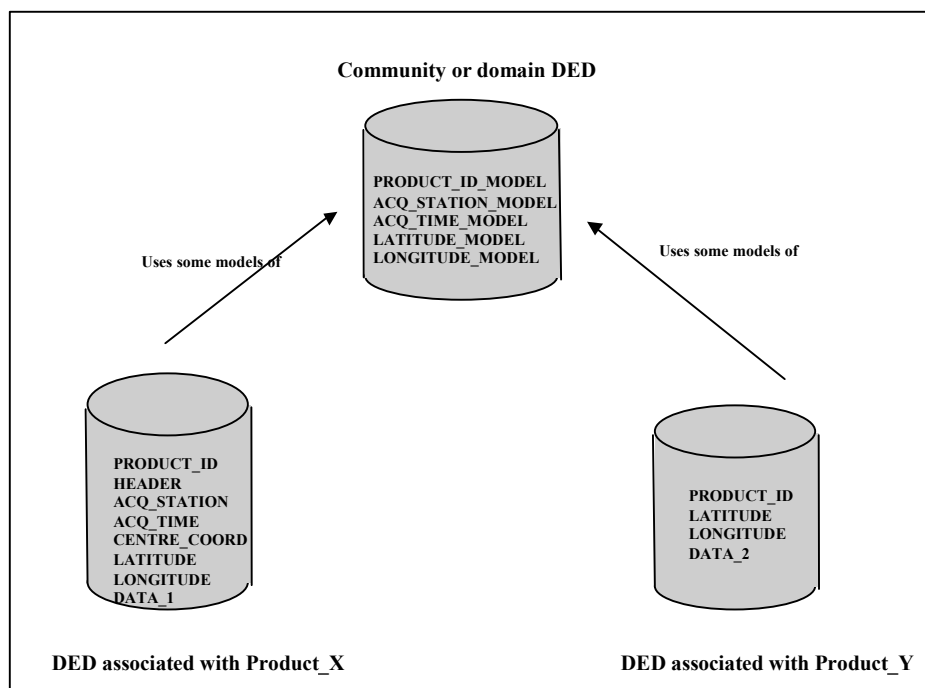


Figure 2-4: ‘Uses some models of’ Links between Data Product Dictionaries

2.3 APPLICATION OF THE DEDSL

2.3.1 GENERAL

As demonstrated in the previous section, there are two major uses for dictionaries:

- to describe a data product semantically;
- to build up and define a community DED.

This Recommendation focuses on developing standard names and descriptions for the concepts required for Data Entity Dictionaries, and formally defines the concepts of name, definition, units, and a small set of other attributes so they may be used consistently in the formation of data entity dictionaries. A method is also provided to permit the set of attributes to be extended beyond the standard ones provided within this Recommendation.

This formal definition enables the definition of generic tools to assist producers in creating documented products, and to assist consumers in understanding the products they receive.

2.3.2 PRODUCT DATA ENTITY DICTIONARIES

A product DED is a means for an organisation to present the semantics necessary for a good understanding of a managed data product. A product DED is dependent of the data product description. However, an enhanced understanding is only possible when the semantics associated with the products are presented in a common, i.e., standardised way.

Therefore, this Recommendation provides a foundation for the creation of a product DED by providing a basic set of concepts for data entity descriptions. This Recommendation also provides the formal methods to describe relationships among the data entities of the product DEDs.

2.3.3 COMMUNITY DATA ENTITY DICTIONARIES

A community DED is also a means for an organisation to gain some degree of control/standardisation over the data descriptions created by member data producers. Unlike product data entity dictionaries, these community DED are not used in conjunction with a specific product description technique. They are independent of the specific implementation of products. Examples of uses of community DED include:

- the creation of a standard data entity dictionary by an organisation that mandates the attributes defining each entity description in dictionaries used within that organisation;
- the creation of a community DED by a particular community (e.g., planetary science, astrophysics, etc.), to establish a degree of standardisation for the contents of any data entity dictionary associated with a data product from that community.

This Recommendation provides a foundation for the creation of community DED and also provides the formal methods to describe relationships among data entities of multiple data entity dictionaries.

2.4 REGISTERING DED

Whenever a project or data designer has defined a product DED, it makes sense to register it as it may apply to multiple instances of the product and this makes it easier to find and retrieve for dissemination or updating.

Whenever a project or data designer has defined a community DED, they can decide to register it at different levels:

- in the framework of any organisation dealing with data of a particular domain or project;
- internally within an agency;
- within the CCSDS community.

For example, whenever a member agency considers that one of its particular community DED corresponds to the needs of other agencies, it may submit its DEDs to an organisation conforming to the CCSDS (references [4] and [C5]) or ISO registration procedures.

3 DESCRIPTORS OF A DATA ENTITY ATTRIBUTE

3.1 GENERAL

3.1.1 The semantic information required for describing a data entity is seen as a collection of attributes. Each attribute describes a particular semantic characteristic of the data entity.

3.1.2 Data entity attributes shall be registered and controlled in a standard way in order to achieve consistency in the exchange of information on data entities among data entity dictionaries, and to enable the comparison of data entities used in different management environments.

3.1.3 Therefore, this Recommendation defines a list of general descriptors for describing a data entity attribute.

3.2 LIST OF DESCRIPTORS

3.2.1 A data entity attribute is specified by means of attribute descriptors.

3.2.2 A descriptor is either mandatory, conditional, optional or defaulted when a data entity attribute is defined:

- **mandatory**: always required;
- **conditional**: required under specified conditions;
- **optional**: allowed but not required;
- **defaulted**: provides a default value for the descriptor when the descriptor is omitted.

3.2.3 A descriptor can only appear once in a data entity attribute description and the descriptor name is not case-sensitive.

NOTE – Table 3-1 provides the set of general descriptors defined by this Recommendation. The obligation column indicates whether a descriptor is mandatory (M), conditional (C), optional (O) or defaulted (D).

Table 3-1: General Descriptors

descriptor of attribute	obligation
ATTRIBUTE_NAME	M
ATTRIBUTE_DEFINITION	M
ATTRIBUTE_OBLIGATION	M
ATTRIBUTE_CONDITION	C
ATTRIBUTE_MAXIMUM_OCCURRENCE	M
ATTRIBUTE_VALUE_TYPE	M
ATTRIBUTE_MAXIMUM_SIZE	O
ATTRIBUTE_ENUMERATION_VALUES	C
ATTRIBUTE_COMMENT	O
ATTRIBUTE_INHERITANCE	D
ATTRIBUTE_DEFAULT_VALUE	C
ATTRIBUTE_VALUE_EXAMPLE	O
ATTRIBUTE_SCOPE	D

3.3 ATTRIBUTE_NAME

Purpose Label assigned to a data entity attribute.

Descriptor Name The standard term to be used for this descriptor is **attribute_name**.

Obligation This descriptor is mandatory.

Descriptor Type The value of this descriptor is of type **Identifier**.

The **attribute_name** shall be unique within a Data Entity Dictionary.

Interoperability The maximum length for the value of this descriptor is 40 characters.

Constraint The Identifier is not case-sensitive.

The Identifier shall not contain any white spaces (e.g., one or more space characters, carriage returns, line feeds, form feeds or tabs).

It shall begin with a letter and may be followed by letters, digits or the underline character.

It shall end with a letter or a digit.

It shall only contain characters in a visibly displayable form.

3.4 ATTRIBUTE_DEFINITION

Purpose The definition is required to give the description of the data entity attribute. This definition is intended for human readership and therefore any information that increases the understanding of the identified attribute should be included.

Descriptor Name The standard term to be used for this descriptor is **attribute_definition**.

Obligation This descriptor is mandatory.

Descriptor Type The value of this descriptor is of type **Text**.

Interoperability Constraint It is intended that the value of this descriptor can be of a significant length and hence provide a description of the attribute as complete as possible.

The maximum length for the value of this descriptor is 8000 characters.

3.5 ATTRIBUTE_OBLIGATION

Purpose Descriptor indicating whether a data entity attribute shall always be present, or only sometimes present, according to specified conditions.

Descriptor Name The standard term to be used for this descriptor is **attribute_obligation**.

Obligation This descriptor is mandatory.

Descriptor Type The value of this descriptor is of type **Enumerated** with values corresponding to the following cases:

- **mandatory**: the data entity attribute shall always be present.
- **conditional**: the data entity attribute shall be present if conditions specified by the descriptor **attribute_condition** occur for the same data entity attribute.
- **optional**: the data entity attribute may be present or not.
- **defaulted**: a data entity attribute that assumes a specified default value if it is omitted from a data entity description. The specified default value is provided by the **attribute_default_value** descriptor. Composite attributes, that is attributes whose `attribute_value_type` is a List or Choice, shall not have an obligation of defaulted.

Interoperability Constraint The coding values are ‘M’ or ‘mandatory’, ‘C’ or ‘conditional’, ‘O’ or ‘optional’ and ‘D’ or ‘defaulted’; these values are not case-sensitive.

3.6 ATTRIBUTE_CONDITION

Purpose Descriptor indicating the circumstances under which a data entity attribute shall be present.

Descriptor Name The standard term to be used for this descriptor is **attribute_condition**.

Obligation This descriptor is conditional.

It shall be present if the **attribute_obligation** descriptor of the same data entity attribute has the value ‘**conditional**’.

Descriptor Type The value is of type **Text**.

Interoperability The maximum length for the value of this descriptor is 8000 characters.

Constraint

It does not exclude the possibility of defining a particular formalism for the text itself, enabling the automatic processing of the text.

3.7 ATTRIBUTE_MAXIMUM_OCCURRENCE

Purpose Descriptor specifying the maximum number of instances which the data entity attribute may have in the specification of a data entity.

Descriptor Name The standard term to be used for this descriptor is **attribute_maximum_occurrence**.

Obligation This descriptor is mandatory.

Descriptor Type The value of this descriptor is of type **Integer**, or of type **Character** with the value of ‘**n**’. The character ‘**n**’ specifies that there is no upper limit on the number of times that the data entity attribute may occur.

Interoperability ‘**n**’ denoting an unlimited number is not case-sensitive.

Constraint

3.8 ATTRIBUTE_VALUE_TYPE

Purpose Descriptor specifying a set of distinct values for representing the attribute value.

Descriptor Name The standard term to be used for this descriptor is **attribute_value_type**.

Obligation This descriptor is mandatory.

Descriptor Type The value of this descriptor can be of type Enumerated with the following values: **Enumerated**, **Integer**, **Real**, **Text**, **Identifier**, **Entity_Type**.

When the value of the descriptor is **Entity_Type** then the attribute value must be an instance of the type of the data entity being defined; this type may be Enumerated, Integer, Real, Text or Composite and cannot be an Identifier.

The descriptor value can also be made up of an ordered list of elementary types or an included list or choice. (Note that in the examples in this document we use parentheses with comma delimited elements to show the elements of the list, indicated by the initial keyword **List**.) When the descriptor value is an ordered list of the same elementary type, it indicates that there may be an indefinite repetition of the same elementary type.

The descriptor value can be specified as a choice among a list of elementary types or an included list or choice. (Note that in the examples in this document we use parentheses with comma delimited elements, to delimit elements in a choice, indicated by the initial keyword **Choice**.)

Note that the following examples show the use of `attribute_value_type` as used in the definition of the standard attributes for data entities that are defined in subsection 4.4.

NOTE – The following example presents an attribute of elementary type:

For the definition of the attribute called **name** that identifies the data entity, the `attribute_value_type` specifies that a value of this attribute is an identifier.
attribute_value_type: Identifier

Example 3-1: Elementary Type

NOTE – The following example presents an attribute type defined as a set of two elementary types:

For the definition of the attribute called **alias** that defines an alternative name of a data entity, the `attribute_value_type` specifies that a value of this attribute is composed of the alternative name (of the type **Text**) followed by the context in which this alias is used (of the type **Text**).
attribute_value_type: List(Text, Text)

Example 3-2: Ordered List of Elementary Types

NOTE – The following example presents an attribute type defined as an indefinite repetition of the same elementary type:

For the definition of the attribute called **enumeration_values** that provides the set of permitted values of data entities of type Enumerated, the `attribute_value_type` specifies that a value of this attribute is an indefinite repetition of the same elementary type which is **Identifier**.
attribute_value_type: List(Identifier)

Example 3-3: Ordered List of the Same Elementary Type

NOTE – The following example illustrates the use of `Entity_Type`:

For the definition of the attribute called **specific_instance** that provides a specific value of the data entity occurrence, the `attribute_value_type` specifies that a value of this attribute is composed by a specific value (of the type **Entity_Type**) followed of the meaning and context of this specific value (of the type **Text**).
attribute_value_type: List(Entity_Type, Text)

Example 3-4: Use of `Entity_Type`

NOTE – The following example illustrates the use of Choice:

For the definition of the attribute called **text_size** that provides a limitation on the size of the values of data entities of type Text, the `attribute_value_type` specifies that a value of this attribute is either the exact size of the text when known (of the type **Integer**) or a list made up of the minimum and the maximum number of characters the text may contain (both being of type **Integer**).
attribute_value_type: Choice (Integer, List(Integer, Integer))

Example 3-5: Use of Choice

3.9 ATTRIBUTE_MAXIMUM_SIZE

Purpose Descriptor specifying the maximum number of characters for representing the value of the attribute.

Descriptor Name The standard term to be used for this descriptor is **attribute_maximum_size**.

Obligation This descriptor is optional.

It shall be present if the **attribute_value_type** value is **Identifier** or **Text**.

It may be present for composite attributes made up of Identifiers or Texts. It then represents the size allowed for each component of the set.

Descriptor Type The value of this descriptor is of type **Integer** when the **attribute_value_type** value is **Identifier** or **Text**.

When the attribute is composite made up of Identifiers or Texts expressed as **List** or **Choice**, the value of this descriptor follows the same formalism as the one defined for the descriptor **attribute_value_type**.

NOTE – The following example illustrates the definition of the **attribute_maximum_size** for a composite attribute.

For the definition of the attribute called **relation** that expresses a relationship between two entity definitions and whose **attribute_value_type** is expressed as '**Choice(List(Text,Identifier), List(Text,Identifier,Identifier))**', **attribute_maximum_size** specifies the size allowed for each component as follows:
attribute_maximum_size : Choice(List(8000, 400), List(8000,400,400))

Example 3-6: Use of Attribute_Maximum_Size

3.10 ATTRIBUTE_ENUMERATION_VALUES

Purpose Descriptor specifying the distinct and discrete values of the attribute.

Descriptor Name The standard term to be used for this descriptor is **attribute_enumeration_values**.

Obligation This descriptor is conditional.

It shall be present if the **attribute_value_type** value is **Enumerated**.

Descriptor Type The value of this descriptor is a comma-separated set of **Identifier** within parentheses without any leading keyword.

Interoperability Constraint The interoperability constraints defined in 3.2 relative to Identifiers also apply here.

NOTE – The following example illustrates the definition of the descriptor:

For the definition of the attribute called **data_type** that provides the type of the data entity occurrence, the **attribute_enumeration_values** specifies the distinct allowed types.
attribute_enumeration_values : (enumerated, text, real, integer, composite)

Example 3-7: Use of Attribute_Enumeration_Values

3.11 ATTRIBUTE_COMMENT

Purpose Descriptor providing information which is not directly required to understand the meaning of the attribute, but which could still assist the user of the attribute in some manner. It may also contain examples.

Descriptor Name The standard term to be used for this descriptor is **attribute_comment**.

Obligation This descriptor is optional.

Descriptor Type The value of this descriptor is a of type **Text**.

Interoperability Constraint The maximum length for the value of this descriptor is 8000 characters.

3.12 ATTRIBUTE_INHERITANCE

Purpose Descriptor providing information about the inheritance rules for the attribute in a context of data entity modeling.

Descriptor Name The standard term to be used for this descriptor is **attribute_inheritance**.

Obligation This descriptor is defaulted.

Descriptor Type The value of this descriptor is of type **Enumerated** with two discrete values: **inheritable** and **not_inheritable**.

The context is as follows: a data entity description A inherits from another data entity description B. The following cases describe what may happen for the values of the attributes of A for the different possible values of attribute_inheritance for the attributes of B.

- When the value of an attribute of B cannot be inherited, the attribute may be defined locally in the description of A.
- When the value of an attribute of B can be inherited, the value of this attribute is the value of the corresponding attribute of A, to which specialization rules have been applied as defined in 4.6.3.

Default value **inheritable**

3.13 ATTRIBUTE_DEFAULT_VALUE

Purpose Descriptor providing a default value for the attribute.

Descriptor Name The standard term to be used for this descriptor is **attribute_default_value**.

Obligation This descriptor is conditional.

This descriptor must be present if and only if the current described data attribute has its **attribute_obligation** descriptor equal to **'defaulted'** except in the case of a composite attribute, that is for an attribute whose **attribute_value_type** is a List, or a Choice, where it must be omitted.

Descriptor Type The format of this descriptor must conform to the type of the attribute that it illustrates, i.e., must be compliant with the value of the descriptor **attribute_value_type**.

3.14 ATTRIBUTE_VALUE_EXAMPLE

Purpose Descriptor providing examples for the value of the attribute.

Descriptor Name The standard term to be used for this descriptor is **attribute_value_example**.

Obligation This descriptor is optional.

Descriptor Type The value of this descriptor is of type **Text**.

Interoperability The text may contain examples and explanatory information.

Constraint Therefore, it can be of a significant length.

The maximum length for the value of this descriptor is 8000 characters.

3.15 ATTRIBUTE_SCOPE

Purpose Descriptor specifying the category of entities in which the attribute may appear.

Descriptor Name The standard term to be used for this descriptor is **attribute_scope**.

Obligation This descriptor is defaulted.

Descriptor Type The value of this descriptor is of type Enumerated with three discrete values: **data**, **dictionary** and **all**.

- **data**: means that the attribute may appear only as a data entity attribute;
- **dictionary**: means that the attribute may appear only as a data entity dictionary attribute and is applicable to the entire collection of data entities in the dictionary;
- **all**: means that the attribute may appear as a data entity attribute as well as a data entity dictionary attribute, in which case the value in the data entity definition takes precedence.

Default value data

Interoperability The coding values are respectively ‘data’, ‘dictionary’ and ‘all’.

Constraint These values are not case-sensitive.

4 DATA ENTITY ATTRIBUTES AND DATA ENTITY DICTIONARY ATTRIBUTES

4.1 CONCEPT DEFINITIONS

4.1.1 A specification or semantic description of a data entity consists of a set of attributes.

4.1.2 Some of these attributes which are considered as frequently needed are defined in this Recommendation as **standard attributes** (see 4.2 through 4.4). Additional attributes called **user-defined attributes** may be required according to the user's needs. The standard attributes are of character type, however a user-defined attribute may be a character pointer which may point to other, non-character, data. The method to define user-defined attributes is described in 4.5.

4.1.3 A **data entity dictionary** is a collection of data entity descriptions. In this view, a dictionary can be seen as a '**dictionary entity**' that will also be described with attributes (called **dictionary attributes**).

4.1.4 Standard attributes will be used to describe:

- some features of the dictionary as a whole (see 4.3);
- the data entities in the dictionary (see 4.4).

4.1.5 Three classes of data entities are defined:

- **model**: a data entity described independently from any instance in a data product (e.g., in a 'domain' dictionary), and corresponding to a re-usable data entity model from which other data entities may inherit the attributes and apply some specialization rules;
- **data field**: a data entity in a data product, having its own specific attributes;
- **constant**: a named constant value used within a dictionary but not being part of the data themselves. Such a class enables data entity dictionaries to specify values which will be used by several projects or within a domain (astronomy constants, image size, etc.).

4.1.6 It is important to make clear in a dictionary what is the class of the defined entities, as these entities will be used differently according to their class, although they will be described with the same attributes.

4.1.7 The standard attributes are described using the descriptors previously defined and can be:

- **mandatory**: always required;
- **conditional**: required under specified conditions;
- **optional**: allowed but not required;

- **defaulted:** provides a default value for the attribute when the attribute is omitted.

4.1.8 These attributes are classified into five categories:

- **identifying attributes** that are applicable for the identification of data or dictionary entities;
- **definitional attributes** that describe core semantic aspects of data or dictionary entities;
- **relational attributes** that describe associations among data or dictionary entities;
- **representational attributes** that describe interpretational aspects of data or dictionary entities;
- **administrative attributes** that describe management and control aspects of dictionary entities.

4.2 GENERAL VIEW OF THE STANDARD ATTRIBUTES

4.2.1 STANDARD ATTRIBUTES FOR DATA ENTITY DICTIONARIES

Table 4-1 provides for each category the standard attributes defined by this Recommendation for a Data Entity Dictionary. They are applicable to the entire collection of data entities in the dictionary. The obligation column indicates whether an attribute is mandatory (M), conditional (C), optional (O) or defaulted (D).

Table 4-1: Data Entity Dictionary Attributes

Attribute Category	Name of data entity attribute	Obligation
Identifying	DICTIONARY_NAME	M
Definitional	DICTIONARY_DEFINITION	O
Relational	EXTERNAL_DICTIONARY_REFERENCE	C
Representational	TEXT_FIELD_CHARACTER_SET	M
	CASE_SENSITIVITY	D
	LANGUAGE	M
Administrative	DICTIONARY_VERSION	O
	DICTIONARY_IDENTIFIER	O
	DEDSL_VERSION	M

4.2.2 STANDARD ATTRIBUTES FOR DATA ENTITIES

Table 4-2 provides gives for each category the standard attributes that are defined by this Recommendation for data entities. The obligation column indicates whether an attribute is mandatory (M), conditional (C), optional (O) or defaulted (D) in the definition of each data entity appearing in a conforming DED.

Table 4-2: Data Entity Attributes

Attribute Category	Name of data entity attribute	Obligation
Identifying	NAME	M
	ALIAS	O
	CLASS	D
Definitional	DEFINITION	M
	SHORT_DEFINITION	O
	COMMENT	O
	UNITS	C
	SPECIFIC_INSTANCE	O
Relational	INHERITS_FROM	O
	COMPONENT	O
	KEYWORD	O
	RELATION	O
Representational	DATA_TYPE	C
	ENUMERATION_VALUES	C
	ENUMERATION_MEANING	O
	ENUMERATION_CONVENTION	O
	RANGE	O
	TEXT_SIZE	C
	CASE_SENSITIVITY	O
	LANGUAGE	O
	CONSTANT_VALUE	C

4.3 STANDARD ATTRIBUTES FOR DATA ENTITY DICTIONARIES

4.3.1 DICTIONARY_NAME

Attribute_Name	: dictionary_name
Attribute_Definition	: Human readable name for the Data Entity Dictionary
Attribute_Obligation	: Mandatory
Attribute_Value_Type	: Identifier
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: dictionary

Interoperability Constraints

Attribute_Maximum_Size	: 400
Attribute_Comment	: The Identifier shall not contain any white spaces (e.g., one or more space characters, carriage returns, line feeds, form feeds or tabs).

It shall begin with a letter and may be followed by letters, digits or the underline character.

It shall end with a letter or a digit.

It shall only contain characters in a visibly displayable form.

4.3.2 DICTIONARY_DEFINITION

Attribute_Name	: dictionary_definition
Attribute_Definition	: Human readable definition for the Data Entity Dictionary
Attribute_Obligation	: Optional
Attribute_Value_Type	: Text
Attribute_Maximum_ Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: dictionary
Attribute_Comment	: The value of this attribute is a free format text which can span a number of lines.

Interoperability Constraint

Attribute_Maximum_Size : 8000

4.3.3 EXTERNAL_DICTIONARY_REFERENCE

Attribute_Name	: external_dictionary_reference
Attribute_Definition	: This attribute gives a reference to another Data Entity Dictionary whose models are reused in the current one. This reference is made up of the following information: <ul style="list-style-type: none"> – the local name to use to refer to the referenced external Data Entity Dictionary—unique within this dictionary; – the dictionary identifier within a registration authority, e.g., an official reference registration (ADID) by a CCSDS Control Authority; – text which identifies the registration authority (NOTE that this is not standardised at the moment).
Attribute_Obligation	: Conditional
Attribute_Value_Type	: List(Identifier , Identifier, Text)
Attribute_Maximum_Occurrence	: ‘n’
Attribute_Condition	: This attribute is mandatory if a reference to a Data Entity Dictionary is made in the current data entity dictionary (in one of the ‘inherits_from’ or ‘relation’ attributes).
Attribute_Value_Example	: (CDPP_Plasma_Dictionary, FCST0172, CCSDS_Control_Authority)
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: dictionary

Interoperability Constraints

Attribute_Maximum_Size	: List (400, 400, 400)
Attribute_Comment	: The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.3.4 TEXT_FIELD_CHARACTER_SET

Attribute_Name	: text_field_character_set
Attribute_Definition	: Name of the Character Set that is valid for TEXT descriptor type and TEXT attribute value type within the dictionary
Attribute_Obligation	: Mandatory
Attribute_Value_Type	: Text
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: dictionary

Interoperability Constraint

Attribute_Maximum_Size : 40

4.3.5 CASE_SENSITIVITY

Attribute_Name	: case_sensitivity
Attribute_Definition	: The value of this attribute specifies the case sensitivity for the Identifiers used as values for the attributes of the current entity. When used in a data entity, the value of the attribute overrides the value specified at the dictionary level.
Attribute_Obligation	: Defaulted
Attribute_Value_Type	: Enumerated
Attribute_Enumeration_Values	: (case_sensitive, not_case_sensitive)
Attribute_Default_Value	: not_case_sensitive
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: all

4.3.6 LANGUAGE

Attribute_Name	: language
Attribute_Definition	: Main natural language that is valid for any value of type TEXT given to the attributes of the current entity. When used in a data entity, the value of the attribute overrides the value specified for the dictionary entity.
Attribute_Obligation	: Mandatory
Attribute_Value_Type	: List(Text, Identifier) where: <ul style="list-style-type: none"> – text is provided in English and corresponds to the English name of the language as specified in ISO 639-2 (see reference [5]); – identifier refers to a 2 or 3 letter country code as specified in ISO 639-2.
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Value_Example	: ('French', fr)
Attribute_Scope	: all

Interoperability Constraints

Attribute_Maximum_Size	: List(40, 3)
Attribute_Comment	: The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.3.7 DICTIONARY_VERSION

Attribute_Name	: dictionary_version
Attribute_Definition	: Version of the Data Entity Dictionary
Attribute_Obligation	: Optional
Attribute_Value_Type	: Text
Attribute_Maximum_ Occurrence	: 1
Attribute_Comment	: It corresponds to the issue and the revision of the current dictionary separated by a period.
Attribute_Value_Example	: '1.a'
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: dictionary

4.3.8 DICTIONARY_IDENTIFIER

Attribute_Name	: dictionary_identifier
Attribute_Definition	: The Identifier under which the Data Entity Dictionary has been registered at a registration Authority.
Attribute_Obligation	: Optional
Attribute_Value_Type	: Identifier
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: dictionary

Interoperability Constraints

Attribute_Maximum_Size	: 400
Attribute_Comment	: If the registration authority is the CCSDS Control Authority (see references [4] and [C-5]), the format of this attribute value is an unquoted ASCII string of eight consecutive Restricted ASCII characters that constitute a registered MACAO ADID. In case of other registration authorities, the attribute value follows the internal registration identification system of the authority.

The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.3.9 DEDSL_VERSION

Attribute_Name	: dedsl_version
Attribute_Definition	: CCSDS document number of the document corresponding to the implementation of the Abstract Syntax. Note that this reference contains the version.
Attribute_Obligation	: Mandatory
Attribute_Value_Type	: Text
Attribute_Maximum_ Occurrence	: 1
Attribute_Value_Example	: ‘CCSDS 647.2-B-1.0’
Attribute_Comment	: For example, a PVL or XML implementation.
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: dictionary

4.4 STANDARD ATTRIBUTES FOR DATA ENTITIES

4.4.1 IDENTIFYING ATTRIBUTES

4.4.1.1 Name

Attribute_Name : **Name**

Attribute_Definition : The value of this attribute may be used to link a collection of attributes with an equivalent identifier in, or associated with, the data entity.

The value of this attribute may also be used by the software developer to name corresponding variables in software code or designate a field to be searched for locating particular data entities.

The **name** shall be unique within a Data Entity Dictionary.

Attribute_Obligation : Mandatory

Attribute_Value_Type : **Identifier**

Attribute_Maximum_
Occurrence : 1

Attribute_Value_Example : ACQ_STATION

Attribute_Inheritance : Not_inheritable

Interoperability Constraints

Attribute_Maximum_Size : 400

Attribute_Comment : The interoperability constraints defined in 4.3.1. relative to Identifiers also apply here.

4.4.1.2 Alias

Attribute_Name : **Alias**

Attribute_Definition : Single- or multi-word designation that differs from the given name, but represents the same data entity concept, followed by the context in which this name is applied.

The value of this attribute provides an alternative designation of the data entity that may be required for the purpose of compatibility with historical data or data deriving from different sources. For example, different sources may produce data including the same entities, but giving them different names. Through the use of this attribute it will be possible to define the semantic information only once. Along with the alternative designation, this attribute value shall provide a description of the context of when the alternative designation is used.

The value of the alternative designation can also be searched when a designation used in a corresponding syntax description is not found within the **NAME** values.

Attribute_Obligation : Optional

Attribute_Value_Type : **List(Text, Text)**

Attribute_Maximum_Occurrence : 'n'

Attribute_Value_Example : ('TIME_LINE', 'used within the ground segment')

Attribute_Inheritance : Not_inheritable

Interoperability Constraint

Attribute_Maximum_Size **List(400, 400)**

4.4.1.3 Class

Attribute_Name	: class
Attribute_Definition	: The value of this attribute makes a clear statement of what kind of entity is defined by the current entity definition. This definition can be a model definition, a data field definition, or a constant definition.
Attribute_Obligation	: Defaulted
Attribute_Value_Type	: Enumerated
Attribute_Enumeration_Values	: (model, data_field, constant)
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Default_Value	: data_field

4.4.2 DEFINITIONAL ATTRIBUTES

4.4.2.1 Definition

Attribute_Name : **definition**

Attribute_Definition : Statement that expresses the essential nature of a data entity and permits its differentiation from all the other data entities.

This attribute is intended for human readership and therefore any information that will increase the understanding of the identified data entity should be included.

It is intended that the value of this attribute can be of significant length and hence provide a description of the data entity as complete as possible. The value of this attribute can be used as a field to be searched for locating particular data entities.

Attribute_Obligation : Mandatory

Attribute_Value_Type : **Text**

Attribute_Maximum_
Occurrence : 1

Attribute_Comment : The value of this attribute may include the same semantic information in natural language as the one carried in a more formal manner by other attributes. This is neither a requirement nor illegal, but the user must make sure that inconsistencies do not arise.

Interoperability Constraint

Attribute_Maximum_Size : 8000

4.4.2.2 Short_Definition

Attribute_Name : **short_definition**

Attribute_Definition : Statement that expresses the essential nature of a data entity in a shorter and more concise manner than the statement of the mandatory attribute: **definition**.

This attribute provides a summary of the more detailed information provided by the **definition** attribute.

The value of this attribute can be used as a field to be searched for locating particular data entities. It is also intended to be used for display purposes by automated software, where the complete definition value would be too long to be presented in a convenient manner to users.

Attribute_Obligation : Optional

Attribute_Value_Type : **Text**

Attribute_Maximum_
Occurrence : 1

Interoperability Constraint

Attribute_Maximum_Size : 80

4.4.2.3 Comment

Attribute_Name	:	comment
Attribute_Definition	:	Associated information about a data entity. It enables to add information which does not correspond to definition information.
Attribute_Obligation	:	Optional
Attribute_Value_Type	:	Text
Attribute_Maximum_ Occurrence	:	'n'

Interoperability Constraint

Attribute_Maximum_Size	:	8000
------------------------	---	------

4.4.2.4 Units

Attribute_Name	: units
Attribute_Definition	: Attribute that specifies the scientific units that should be associated with the value of the data entity so as to make the value meaningful to applications.
Attribute_Obligation	: Conditional
Attribute_Condition	: If the data entity is non-scalar then the attribute shall not be specified. If the data entity is of a scientific scalar type (Integer or Real) then it can appear several times for data entities of class model and only once for data entities of class data_field or constant .
Attribute_Value_Type	: Text <p>If the scalar type has no unit, e.g., a ratio, then the UNITS attribute should be given a particular value (e.g., NO_UNIT). It is up to the implementation recommendation to define this particular value.</p> <p>The contents of the text must conform to the representation specified in ISO 2955 (see reference [C-10]).</p> <p>As detailed in ISO 2955, the following conventions apply when combining units:</p> <ul style="list-style-type: none"> – multiplication shall be indicated by a period (.), e.g., Pa . s to designate Pascal second, the unit of dynamic viscosity; – negative exponents shall be indicated by following the unit directly with the numeric power preceded by a minus sign, e.g., m⁻³ to designate m⁻³; – Division shall be indicated by a solidus (/), e.g., m/s, or by expressing the denominator with a negative exponent, e.g., m . s⁻¹; – Positive exponents shall be indicated by following the unit directly with the numeric power with no sign, e.g., m² to designate m² <p>Decimal multiples of units shall be indicated by the combination of a prefix representation (see reference [C9]) immediately before the unit, e.g., kN to represent kilo Newtons.</p>
Attribute_Maximum_Occurrence	: 'n'

Interoperability Constraint

Attribute_Maximum_Size : 80.

4.4.2.5 Specific_Instance

Attribute_Name	: specific_instance
Attribute_Definition	: Attribute that provides a real-world meaning for a specific instance (a value) of the data entity being described. The reason for providing this information is so that the user can see that there is some specific meaning associated with a particular value instance that indicates something more than just the abstract value. For example, the fact that zero degree latitude is the equator could be defined. This means that the value of this attribute must provide both an instance of the entity value and a definition of its specific meaning.
Attribute_Obligation	: Optional
Attribute_Value_Type	: List(Entity_Type, Text) There shall be two values associated with this attribute: an instance value (a literal or a constant name) and a specific meaning definition.
Attribute_Maximum_Occurrence	: 'n'
Attribute_Comment	: The values of the attribute can be used to enhance user interfaces and, therefore, user understanding.
Attribute_Value_Example	: A specific value of DEGREE (with the Entity_Type of Real), could be: (0.0, 'Greenwich').
Attribute_Inheritance	: Not_inheritable

Interoperability Constraint

Attribute_Maximum_Size : **List(80, 400)**

4.4.3 RELATIONAL ATTRIBUTES**4.4.3.1 Inherits_From**

Attribute_Name : **inherits_from**

Attribute_Definition : Gives the name of a model or data field from which the current entity description inherits attributes. This name must be the value of the **name** attribute found in the referred entity description.

Referencing this data entity description means that all the values of its attributes having their **attribute_inheritance** set to **inheritable** apply to the current description.

Attribute_Obligation : Optional

Attribute_Value_Type : **Choice(Identifier , List(Identifier , Identifier))**

The first identifier refers to the name of a model or data field.

The second one gives the local name of the external Data Entity Dictionary where the referred entity description is defined.

The absence of the second identifier assumes that this referred model or data field is local to the current Data Entity Dictionary or, if not, that it is to be found in the dictionary referenced through the **external_dictionary_reference** attribute, which must be unique in that case.

Attribute_Maximum_Occurrence : 1

Attribute_Value_Example : (CCSDS_calendar_time, CCSDS_TIME_CODES)

Attribute_Inheritance : Not_inheritable

Interoperability Constraints

Attribute_Maximum_Size : **Choice(400,List(400, 400))**

Attribute_Comment : This attribute is intended to enable reuse. Each data entity description referring to the same entity should be qualified using the same value of this attribute.

The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.4.3.2 Component

Attribute_Name : **component**

Attribute_Definition : Name of a component, followed by the number of times it occurs in the composite data entity. The number of times is specified by a range.

Attribute_Obligation : Optional

Attribute_Value_Type : **Choice(Identifier, Identifier(a .. b))**

Where a is the minimum number of times the component occurs and b is the maximum number of times it occurs.

a and b are integer literals or constant names.

The following convention applies: the character 'n' indicates that there is no upper limit.

Attribute_Maximum_Occurrence : 'n'

Interoperability Constraints

Attribute_Maximum_Size : 400

Attribute_Comment : This attribute can be used only for composite data entities (arrays, records or lists). When a composite data entity has no component defined, it means that they are not known yet, and will be specified later.

The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.4.3.3 Keyword

Attribute_Name	: keyword
Attribute_Definition	: A significant word used for retrieving data entities
Attribute_Obligation	: Optional
Attribute_Value_Type	: Text
Attribute_Maximum_ Occurrence	: 'n'

Interoperability Constraints

Attribute_Comment	: This attribute can be used for recording keywords (search keys) associated with the data entity in question. Indeed, it enables the possibility of defining a particular formalism for the text, and then an automatic processing of the text according to the needs.
Attribute_Maximum_Size	: 80

4.4.3.4 Relation

Attribute_Name	: relation
Attribute_Definition	: This attribute is to be used to express a relationship between two entity definitions when this relation cannot be expressed using a precise standard relational attribute. In that case the relationship is user-defined and expressed using free text.
Attribute_Obligation	: Optional
Attribute_Value_Type	: Choice(List(Text, Identifier), List(Text, Identifier, Identifier)) where: <ul style="list-style-type: none"> – text provides the reader with the kind of relation that links the two related entities; – the first Identifier is the name of the entity in relation with the one being defined; – the second Identifier gives the local name of the dictionary when this entity is described in an external Data Entity Dictionary.
Attribute_Maximum_Occurrence	: ‘n’
Attribute_Value_Example	: For example, the fact that a constant data entity is being defined that corresponds to the number of pixels of an image may be expressed as a relationship between that constant and the image (entity W_Image that is defined in the Spacecraft_Dictionary dictionary) i.e.: (‘size of’, W_Image, Spacecraft_Dictionary)

Interoperability Constraints

Attribute_Maximum_Size	: Choice(List(8000, 400), List(8000, 400, 400))
Attribute_Comment	: The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.4.4 REPRESENTATIONAL ATTRIBUTES

4.4.4.1 Data_Type

Attribute_Name	: data_type
Attribute_Definition	: It specifies the type of the data entity values. This attribute shall have one of the following values: Enumerated, Text, Real, Integer, Composite .
Attribute_Obligation	: Conditional
Attribute_Condition	: This attribute must be present for a product data field definition and for a constant definition (class attribute set to data_field or constant) and is optional for a model definition (class attribute set to model).
Attribute_Value_Type	: Enumerated
Attribute_Enumeration_Values	: (Enumerated, Text, Real, Integer, Composite)
Attribute_Maximum_Occurrence	: 1
Attribute_Comment	: This attribute defines the conceptual data type of the entity, it is not intended for specifying the physical representation of the entity.

For example, an entity may be defined as an Integer; physically it may be encoded as a 16-bit 2's complement binary number or as an ASCII encoded decimal, but in both cases the **data_type** would be **Integer**.

4.4.4.2 Enumeration Attributes

4.4.4.2.1 General

When a data entity belongs to an enumerated type (attribute **data_type** set to **Enumerated**), it means that each possible value of the enumeration has:

- an enumerated value (an identifier as meaningful as possible);
- a meaning (which can be expressed in free text);
- a convention value (corresponding to a human interpretable value for the coding of the enumeration value).

4.4.4.2.2 Enumeration_Values

Attribute_Name	: enumeration_values
Attribute_Definition	: The set of permitted values of the enumerated data entity.
Attribute_Obligation	: Conditional
Attribute_Condition	: This attribute is mandatory if the data_type is Enumerated . It is not applicable in any other case.
Attribute_Value_Type	: List(Identifier)
Attribute_Value_Example	: (NOMINAL, CALIBRATION, OFF)
Attribute_Maximum_Occurrence	: 1

Interoperability Constraint

Attribute_Comment : The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.4.4.2.3 Enumeration_Meaning

Attribute_Name	: enumeration_meaning
Attribute_Definition	: Enables to give a meaning to each value given by the attribute enumeration_values.
Attribute_Obligation	: Optional
Attribute_Condition	: This attribute is permitted if the data_type is Enumerated . It is not applicable in any other case.
Attribute_Value_Type	: List(List(Identifier,Text))
Attribute_Value_Example	: ((NOMINAL, 'equipment on and working'), (CALIBRATION , 'equipment under calibration tests'), (OFF , 'equipment off'))
Attribute_Maximum_Occurrence	: 1

Interoperability Constraint

Attribute_Comment	: The identifiers must be identical to the values given in the attribute ENUMERATION_VALUES. The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.
-------------------	--

4.4.4.2.4 Enumeration_Convention

Attribute_Name	: enumeration_convention
Attribute_Definition	: Gives guidance on the correspondence between the enumeration_values and the numeric or textual values found within the products.
Attribute_Obligation	: Optional
Attribute_Condition	: This attribute is permitted if the data_type is Enumerated . It is not applicable in any other case.
Attribute_Value_Type	: List(List(Identifier,Text))
Attribute_Value_Example	: ((NOMINAL, '0'), (CALIBRATION , '1'), (OFF , '2'))
Attribute_Maximum_Occurrence	: 1

Interoperability Constraint

Attribute_Comment	: The identifiers must be identical to the values given in the attribute ENUMERATION_VALUES. The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.
-------------------	--

4.4.4.3 Range

Attribute_Name	: range
Attribute_Definition	: The minimum bound and the maximum bound of an Integer or Real data entity
Attribute_Obligation	: Optional
Attribute_Value_Type	: List(Entity_Type, Entity_Type) The first specified value is the minimum bound (literal value or constant name), while the second one is the maximum bound (literal value or constant name).
Attribute_Maximum_Occurrence	: 1
Attribute_Comment	: This attribute only applies for Integer and Real data entities.
Attribute_Value_Example	: For a Real data entity called DEGREE, the range could be as follows: (0.0, 360.0).

4.4.4.4 Text_Size

Attribute_Name	: text_size
Attribute_Definition	: The limitation on the size of the values of a Text data entity. This attribute specifies the minimum and the maximum number of characters the text may contain. If the minimum and the maximum are equal, then this implies that the exact size of the text is known.
Attribute_Obligation	: Conditional
Attribute_Condition	: This attribute is mandatory if the data_type is Text .
Attribute_Value_Type	: Choice(Integer,List(Integer, Integer)) A single integer indicates the maximum only. A List of two integers denotes the minimum and maximum, in that order. The integer(s) can be expressed as a literal value or a constant name.
Attribute_Maximum_Occurrence	: 1

Interoperability Constraint

Attribute_Comment	: The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here for constant names.
-------------------	---

4.4.4.5 Case_Sensitivity

Attribute_Name	: case_sensitivity
Attribute_Definition	: The value of this attribute specifies the case sensitivity for the Identifiers used as values for the attributes of the current entity. When used in a data entity, the value of the attribute overrides the value specified at the dictionary level.
Attribute_Obligation	: Optional
Attribute_Value_Type	: Enumerated
Attribute_Enumeration_Values	: (case_sensitive, not_case_sensitive)
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Scope	: data

4.4.4.6 Language

Attribute_Name	: language
Attribute_Definition	: Main natural language that is valid for any value of type TEXT given to the attributes of the current entity. When used in a data entity, the value of the attribute overrides the value specified for the dictionary entity.
Attribute_Obligation	: Optional
Attribute_Value_Type	: List(Text, Identifier) where: <ul style="list-style-type: none"> – text is provided in English and corresponds to the English name of the language as specified in ISO 639-2 (see reference [5]); – identifier refers to a 2 or 3 letter country code as specified in ISO 639-2.
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable
Attribute_Value_Example	: ('French', fr)
Attribute_Scope	: data

Interoperability Constraints

Attribute_Maximum_Size	: List(40, 3)
Attribute_Comment	: The interoperability constraints defined in 4.3.1 relative to Identifiers also apply here.

4.4.4.7 Constant_Value

Attribute_Name	: constant_value
Attribute_Definition	: The value of this attribute is the value given to a constant (entity whose class attribute is set to constant).
Attribute_Obligation	: Conditional
Attribute_Condition	: This attribute is mandatory if class attribute is set to constant . It is not applicable in any other case.
Attribute_Value_Type	: Entity_Type
Attribute_Maximum_Occurrence	: 1
Attribute_Inheritance	: Not_inheritable

4.5 USER-DEFINED ATTRIBUTES

4.5.1 GENERAL

NOTE – The standard attributes specified in section 4 are those predefined by this Recommendation and must be recognized by any system that states conformance to this Recommendation. It is recognized that there may be further attributes that are more specific to a particular domain, mission or project.

4.5.1.1 This section defines the mechanism used to define specific attributes; then these attributes can be used in the same manner as the standard ones to define the semantics of particular data entities. These attributes shall be called ‘**user-defined attributes**’.

4.5.1.2 User-defined attributes shall not have the same name as any standard attribute.

4.5.2 MECHANISM FOR NEW ATTRIBUTE DEFINITIONS

4.5.2.1 A **User-Defined Attribute** is defined using the attribute descriptors specified in section 3. The description of a User-Defined Attribute follows the same rules as the one of a standard attribute.

4.5.2.2 The mandatory descriptors are:

- Attribute_Name;
- Attribute_Definition;
- Attribute_Obligation;
- Attribute_Value_Type;
- Attribute_Maximum_Occurrence.

4.5.2.3 The conditional descriptors are:

- Attribute_Condition;
- Attribute_Enumeration_Values;
- Attribute_Default_Value.

4.5.2.4 The optional descriptors are:

- Attribute_Maximum_Size;
- Attribute_Value_Example;
- Attribute_Comment.

4.5.2.5 The defaulted descriptors are:

- Attribute_Inheritance;
- Attribute_Scope.

NOTE – The following example gives the definition of the user-defined attribute ASSOCIATED_UTILITIES providing the name of the utility which processes the physical values of a specific data entity within a data product.

Attribute_Name	: ASSOCIATED_UTILITIES
Attribute_Definition	: ‘Attribute that provides the name of the utility used to process the occurrences of the data entity with which the attribute is associated.’
Attribute_Obligation	: Optional
Attribute_Value_Type	: List(Text, Text)
Attribute_Comment	: ‘The first text gives the name of the utility, and the second one specifies the application context of the utility.’

Example 4-1: Defining User-Defined Attribute

NOTE – This second example shows how to define an external reference.

Attribute_Name	: AUDIO_EXAMPLE
Attribute_Definition	: ‘The AUDIO_EXAMPLE attribute is a pointer to a sound file in .wav format providing an example to be heard’.
Attribute_Obligation	: Optional
Attribute_Value_Type	: Text
Attribute_Comment	: ‘The value is expressed as a path name’.

Example 4-2: Defining User-Defined Attribute with an External Reference

NOTE – The attribute usage in a data entity description would be as follows:

NAME	: MUSIC_INSTRUMENT
CLASS	: Model
DEFINITION	: ‘It corresponds to an instrument’.
DATA_TYPE	: Enumerated
ENUMERATION_VALUES	: (Piano, Guitar, Violin, Saxophone)
AUDIO_EXAMPLE	: ‘/home/MUSIC/INSTRUMENTS/SOUND/Piano.wav’

Example 4-3: Usage of a User-Defined Attribute

4.5.3 REGISTRATION OF USER-DEFINED ATTRIBUTES

4.5.3.1 So as to obtain maximum reuse and hence interoperability across missions, projects and agencies, it is desirable that new user-defined attributes which are created by projects are submitted for central registration. This means that they can be reused by other projects, eventually leading to a uniform data entity dictionary across many missions and projects.

4.5.3.2 The advantage of sharing the data entity dictionary is that software can be developed to handle the entity data descriptions, which can then be reused by many other projects. In addition, labeling and identification information appearing on products coming from different projects will be comparable.

4.5.3.3 To register user-defined attributes, the data description registration capabilities detailed in the CCSDS Recommendations on Control Authorities (see references [4] and [C-5]) should be followed.

4.5.3.4 When a user registers a user-defined attribute, the following information must be included:

- identification of the user—this information shall be as defined in the registration of data descriptions with the CCSDS Control Authority (see references [4] and [C-5]);
- a specification of the user-defined attribute using the descriptors defined in section 3;
- if software is available to support processing of the value of the user-defined attribute, this should be submitted with the definition, as shown in the previous example for the user-defined attribute ASSOCIATED_UTILITIES.

4.6 RELATIONSHIP RULES

4.6.1 REFERENCE TO AN EXTERNAL DATA ENTITY DICTIONARY

4.6.1.1 As seen previously (in attributes definition), some Data Entity Dictionary references can be made to other Data Entity Dictionaries.

4.6.1.2 In a Data Entity Dictionary definition, the **external_dictionary_reference** attribute can be used (1 to n times, where n means an unspecified number of times) to reference the data entity dictionaries whose data entities are reused in the current one, or whose data entities are referred to in relationships with the current one.

4.6.1.3 When a data entity is reused (as underlined by the use of the **inherits_from** attribute), the origin of this data entity can be given by the second value of this attribute, which matches the local name given in the **external_dictionary_reference**. If this second value is omitted, it means that the description of the reused data entity is assumed to be local to the current data entity dictionary.

4.6.2 COMPOSITION RELATION

4.6.2.1 As seen previously a data entity may be composite, that is, it can be made up of a series of other data entities. This notion is rendered by the composition relation for which the **component** attribute is defined.

4.6.2.2 The order of the components in data entities is not significant.

4.6.2.3 When a data entity is considered as being composite, it is not obligatory to mention all its component data entities if these are not known yet (in particular, when the dictionary is undergoing a definition process). However, when the data entity dictionary corresponds to a physical data product, the components have to be defined.

4.6.3 INHERITANCE

4.6.3.1 General

4.6.3.1.1 A data entity defined with the **class** attribute set to **model** or **data field** can be reused in other data entity descriptions. The **inherits_from** attribute must then be used in the data entity descriptions which wish to reuse that data entity description.

4.6.3.1.2 Only a **model** may be inherited by other **models**.

4.6.3.1.3 Once a reference has been made to the referred data entity, the current description will then inherit the values of all the attributes of the referred entity which have been defined as **inheritable**. Moreover, specialization rules apply to inheritable attributes.

4.6.3.1.4 All the identifying attributes of data entities and all the dictionary attributes as well as the attributes **inherits_from** and **specific_instance** are **not_inheritable**. These attributes have to be defined locally in the current data entity or dictionary entity.

4.6.3.2 Specialization Rules for the Inheritable Attributes

4.6.3.2.1 Specialization of Optional Attributes

Optional attributes may appear a specified number of times. When an entity inherits from another entity, optional attributes can be present or not in the referred entity, and they can be added in the current entity as long as the number of allowed occurrences is not exceeded.

4.6.3.2.2 Specialization Rules for Various Data_Types

4.6.3.2.2.1 Attributes specified in the referred entity may be specialized but not overridden with a different value. The specialization operations allowed for each attribute is based on the descriptor **ATTRIBUTE_VALUE_TYPE**. It should be noted that no change to an attribute is always considered as a valid specialization.

4.6.3.2.2.2 Values of attributes of type **Integer** or **Real** can be specialized by limiting the range of the value domain defined at the referred entity.

4.6.3.2.2.3 Values of attributes of type **Identifier** cannot be specialized; they must be identical to the value given in the referred entity.

4.6.3.2.2.4 Values of attributes of type **Text** can be specialized by adding text (no deletions) to the value given in the referred entity.

4.6.3.2.2.5 Values of attributes of type **Enumerated** can be specialized by removing some **enumeration_values** and the corresponding **enumeration_meaning** and **enumeration_convention**.

4.6.3.2.2.6 Values of attributes of type **Entity_Type** cannot be specialized and must be identical to the value given in the referred entity.

4.6.3.2.2.7 Values of attributes of type **Choice** can be specialized by removing some elementary types from the set of elementary types of the referred entity.

4.6.3.2.2.8 Value of attributes of type **List** cannot be specialized.

4.6.3.2.3 Specialization of the Definitional Attributes

4.6.3.2.3.1 The attributes **DEFINITION**, **SHORT_DEFINITION** and **COMMENT** can be specialized locally so long as the value enriches the information expressed at the referred entity.

4.6.3.2.3.2 In the case of a **data_field/constant/model** inheriting from a model, the following rules apply for the attribute UNITS:

- if the model has no units attribute, the units attribute of the **data_field/constant/model** is unconstrained;
- if the model provides units attribute values, the units attribute value of the **data_field/constant/model** must be one of those listed in the model.

4.6.3.2.4 Specialization of the Representational Attributes

4.6.3.2.4.1 A **model** does not need to have representational attributes specified. A **data_field** must have them specified.

4.6.3.2.4.2 If a **data_field** inherits from a **model** which does not have them specified, then it must specify them locally.

4.6.3.2.4.3 If a **data_field** inherits from a **model** or a **data_field** which does have them specified, then the above rules of specialization apply.

4.6.3.2.5 Specialization of the Relational Attributes

4.6.3.2.5.1 Relational attributes are inheritable except for the ‘INHERITS_FROM’ attribute. They can be defined locally and their value enriches the information expressed in the referred entity, since their ATTRIBUTE_VALUE_TYPE is Text.

4.6.3.2.5.2 As seen in 4.4.3.2., the COMPONENT attribute is inheritable. This attribute can appear any number of times, therefore components can be added to specialize the definition of the data entity with regard to the referred entity.

4.6.4 OTHER KINDS OF RELATION

In addition to the relationships handled with a specific attribute (see above), some other ones have to be described in data entity dictionaries. In that case, there are two possible ways to proceed:

- define a precise user-defined attribute with appropriate descriptors and use it;
- use the generic standard relation attribute name ‘relation’ and express the relationship in free text.

5 IMPLEMENTATION GUIDELINES

NOTE – This section provides some guidelines concerning the use of attributes, whether ‘standard’, or ‘user-defined’.

5.1 For any single semantic entity description, the standard attributes can be presented or accessed in any order; however, it is recommended that the following order should be used whenever possible so as to present a common style to all users. The mandatory attributes are indicated in bold characters, while the optional and conditional attributes are in italic characters:

NAME
ALIAS
CLASS
DEFINITION
SHORT_DEFINITION
COMMENT
UNITS
SPECIFIC_INSTANCE
INHERITS_FROM
COMPONENT
KEYWORD
RELATION
DATA_TYPE
ENUMERATION_VALUES
ENUMERATION_MEANING
ENUMERATION_CONVENTION
RANGE
TEXT_SIZE
CASE_SENSITIVITY
LANGUAGE
CONSTANT_VALUE

5.2 User-defined attributes should be placed following the standard attributes.

5.3 For any single user-defined attribute definition, the descriptors defining the user-defined attribute can be presented or accessed in any order; however, it is recommended that the following order should be used whenever possible so as to present a common style to all users. The mandatory descriptors are indicated in bold characters, while the optional and conditional descriptors are in italic characters:

ATTRIBUTE_NAME
ATTRIBUTE_DEFINITION
ATTRIBUTE_OBLIGATION
ATTRIBUTE_CONDITION
ATTRIBUTE_MAXIMUM_OCCURRENCE
ATTRIBUTE_VALUE_TYPE
ATTRIBUTE_MAXIMUM_SIZE
ATTRIBUTE_ENUMERATION_VALUES
ATTRIBUTE_COMMENT
ATTRIBUTE_INHERITANCE
ATTRIBUTE_DEFAULT_VALUE
ATTRIBUTE_VALUE_EXAMPLE
ATTRIBUTE_SCOPE

5.4 The attributes for each entity description (or descriptors for each attribute definition) must be grouped in some manner so as to keep them separate from the attributes of other entity descriptions. The methodology for grouping the attributes must be defined formally in the DEDSL implementation syntax.

5.5 It is strongly recommended to define a maximum length for identifiers and texts for readability reasons, as well as for possible automatic processing of the DED.

5.6 Although the standard attributes for defining the dictionary can be presented or accessed in any order, it is recommended that the following order should be used whenever possible so as to present a common style to all users. The mandatory descriptors are indicated in bold characters, while the optional and conditional attributes are in italic characters:

DICTIONARY_NAME
DICTIONARY_DEFINITION
EXTERNAL_DICTIONARY_REFERENCE
TEXT_FIELD_CHARACTER_SET
CASE_SENSITIVITY
LANGUAGE
DICTIONARY_VERSION
DICTIONARY_IDENTIFIER
DEDSL_VERSION

6 DEDSL CONFORMANCE: ABSTRACT DEDSL (ADID = CCSD0011)

6.1 GENERAL

6.1.1 This DEDSL specification (sections 3 and 4) is version 1.0 of the DEDSL specification Recommendation and defines a set of standard attributes by name, with restrictions on their permitted values. Note that this part of the specification does not specify how the attribute names and values are to be linked to any given data object. This allows a variety of formatting approaches to be used for this linking.

6.1.2 A certain number of interoperability constraints appear within this Recommendation. These constraints will increase the likelihood of interoperability between different implementations, resulting in two levels of DEDSL conformance.

6.2 CONFORMANCE LEVEL 1: BASE COMPLIANCE

Implementations which implement all of sections 3 and 4 of this Recommendation except the interoperability constraints will be base compliant with this Recommendation.

6.3 CONFORMANCE LEVEL 2: FULL COMPLIANCE

Implementations which implement all of sections 3 and 4 of this Recommendation will be fully compliant with this Recommendation.

ANNEX A**DEDSL EXAMPLES**

(This annex **is not** part of the Recommendation)

In this section a community DED is presented showing the semantic information relative to the data entities chosen as models. Then the definition of a product DED is given using this community DED for the definition of some of its data entities.

A1 COMMUNITY DED**a) Data Entity Dictionary attributes**

DICTIONARY_NAME	: Planetary_Science_Data_Dictionary
DICTIONARY_DEFINITION	: ‘This dictionary contains data entity definitions relative to planetary science and which may be re-used for defining data products.’
TEXT_FIELD_CHARACTER_SET	: ‘ISO-LATIN ALPHABET No1’
CASE_SENSITIVITY	: NOT_CASE_SENSITIVE
LANGUAGE	: (‘English’, en)
DICTIONARY_VERSION	: ‘1.a’
DEDSL_VERSION	: ‘CCSDS 647.2-B-1.0’

b) Data Entities

NAME	: LATITUDE_MODEL
ALIAS	: ('LAT', 'Used by the historical projects EARTH_PLANET')
CLASS	: MODEL
DEFINITION	: 'Latitudes north of the equator shall be designated by the use of the plus (+) sign, while latitudes south of the equator shall be designated by the use of the minus sign (-). The equator shall be designated by the use of the plus sign (+).'
SHORT_DEFINITION	: 'Latitude'
UNITS	: Deg
SPECIFIC_INSTANCE	: (+00.000, 'Equator')
DATA_TYPE	: REAL
RANGE	: (-90.00, +90.00)

NAME : LONGITUDE_MODEL

ALIAS : ('LON', 'Used by the historical projects EARTH_PLANET')

CLASS : MODEL

DEFINITION : 'Longitudes east of Greenwich shall be designated by the use of the plus sign (+), while longitudes west of Greenwich shall be designated by the use of the minus sign (-). The Prime Meridian shall be designated by the use of the plus sign (+). The 180th meridian shall be designated by the use of the minus sign (-).'

SHORT_DEFINITION : 'Longitude'

UNITS : Deg

SPECIFIC_INSTANCE : (-180.000, 'The 180th Meridian')

DATA_TYPE : REAL

RANGE : (-180.00, +180.00)

NAME : PRODUCT_ID_MODEL

ALIAS : ('PRODUCT_NAME', 'Used by the historical projects EARTH_PLANET to identify their data products')

CLASS : MODEL

DEFINITION : 'The PRODUCT_ID represents a permanent, unique identifier assigned to a data product by its producer'.

SHORT_DEFINITION : 'Product Identification'

DATA_TYPE : TEXT

TEXT_SIZE : 40

A2 DATA ENTITY DICTIONARY ASSOCIATED WITH PRODUCT_X

The models of LATITUDE_MODEL, LONGITUDE_MODEL and PRODUCT_ID_MODEL match the data entities appearing within the data product PRODUCT_X and therefore they are referenced within the current data entity dictionary.

a) Data Entity Dictionary attributes

DICTIONARY_NAME	: PRODUCT_X_Dictionary
DICTIONARY_DEFINITION	: ‘This dictionary contains the data entity definitions relative to the data product PRODUCT_X.’
EXTERNAL_DICTIONARY_REFERENCE	: (Planetary_Science_Data_Dictionary)
TEXT_FIELD_CHARACTER_SET	: ‘ISO-LATIN ALPHABET No1’
CASE_SENSITIVITY	: NOT_CASE_SENSITIVE
LANGUAGE	: (‘English’, en)
DICTIONARY_VERSION	: ‘1.a’
DEDSL_VERSION	: ‘CCSDS 647.2-B-1.0’

b) Data Entities

NAME	: HEADER
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents the header of the data product PRODUCT_X. It identifies an aggregation of values which are associated with an image array.’
SHORT_DEFINITION	: ‘Image Header Values’
COMPONENT	: PRODUCT_ID_X(1 .. 1)
COMPONENT	: ACQ_STATION(1 .. 1)
COMPONENT	: ACQ_TIME(1 .. 1)
COMPONENT	: CENTRE_COORD(1 .. 1)
DATA_TYPE	: COMPOSITE
NAME	: PRODUCT_ID
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents a permanent, unique identifier assigned to the data product PRODUCT_X.’
SHORT_DEFINITION	: ‘Product Identification’
INHERITS_FROM	: PRODUCT_ID_MODEL

NAME : ACQ_STATION
 ALIAS : ('ACQUSTAT', 'used in the FITS header')
 CLASS : DATA_FIELD
 DEFINITION : 'It includes the identifier of the station which has acquired the data'.
 SHORT_DEFINITION : 'Identifier of the acquisition station'
 DATA_TYPE : Enumerated
 ENUMERATION_VALUES : (AMERICA, EUROPE, ASIA)
 ENUMERATION_MEANING : ((AMERICA, 'station located in America'),
 (EUROPE , 'station located in Europe'),
 (ASIA , 'station located in Asia'))

NAME : ACQ_TIME
 ALIAS : ('ACQUTIME', 'Used in the FITS header')
 CLASS : DATA_FIELD
 DEFINITION : 'It represents the date and time of the acquisition of the data. Its format is the following one: YYYY-MM-DDThh:mm:ss.d_>Z. It conforms to the CCSDS ISO rules for date/time definitions. The acquisition time should correspond to the first scan line of the data.'
 SHORT_DEFINITION : 'Date/Time of the data acquisition'
 DATA_TYPE : Text
 TEXT_SIZE : 40

NAME	: CENTRE_COORD
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents a coordinate centre’.
SHORT_DEFINITION	: ‘Centre coordinates’
COMPONENT	: LATITUDE (1 .. 1)
COMPONENT	: LONGITUDE (1 .. 1)
KEYWORDS	: ‘LATITUDE BY LONGITUDE COORDINATE CENTRE’
DATA_TYPE	: COMPOSITE
NAME	: LATITUDE
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents the latitude used for the center coordinate’.
INHERITS_FROM	: LATITUDE_MODEL
NAME	: LONGITUDE
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents the longitude used for the centre coordinate’.
INHERITS_FROM	: LONGITUDE_MODEL

NAME : W_IMAGE_SIZE
 CLASS : CONSTANT
 DEFINITION : ‘It represents the number of pixels for an image taken from spacecraft W’.
 SHORT_DEFINITION : ‘Spacecraft W Image pixel’
 RELATION : (‘size of’, DATA_1)
 DATA_TYPE : INTEGER
 CONSTANT_VALUE : 1 440 000

NAME : DATA_1
 CLASS : DATA_FIELD
 DEFINITION : ‘It represents an image taken from spacecraft W’.
 SHORT_DEFINITION : ‘Spacecraft W Image’
 COMMENT : ‘The image is an array of W_IMAGE_SIZE items called DATA_1_PIXEL’
 COMPONENT : DATA_1_PIXEL (1 .. W_IMAGE_SIZE)
 KEYWORD : ‘IMAGE’
 DATA_TYPE : COMPOSITE

NAME : DATA_1_PIXEL
 CLASS : DATA_FIELD
 DEFINITION : ‘It represents a pixel belonging to the image taken from spacecraft W’.
 SHORT_DEFINITION : ‘Spacecraft W Image pixel’
 DATA_TYPE : INTEGER
 RANGE : (0 , 255)

A3 DATA ENTITY DICTIONARY ASSOCIATED WITH PRODUCT_Y

The models of LATITUDE_MODEL, LONGITUDE_MODEL and PRODUCT_ID_MODEL match the data entities appearing within the data product PRODUCT_Y and, therefore, they are referenced within the current data entity dictionary.

a) Data Entity Dictionary attributes

DICTIONARY_NAME	: PRODUCT_Y_Dictionary
DICTIONARY_DEFINITION	: ‘This dictionary contains the data entity definitions relative to the data product PRODUCT_Y’.
EXTERNAL_DICTIONARY_REFERENCE	: (Planetary_Science_Data_Dictionary)
TEXT_FIELD_CHARACTER_SET	: ‘ISO-LATIN ALPHABET No1’
CASE_SENSITIVITY	: NOT_CASE_SENSITIVE
LANGUAGE	: (‘English’, en)
DICTIONARY_VERSION	: ‘1.a’
DEDSL_VERSION	: ‘CCSDS 647.2-B-1.0’

b) Data Entities

NAME	: PRODUCT_ID
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents a permanent, unique identifier assigned to the data product PRODUCT_Y’.
SHORT_DEFINITION	: ‘Product Identification’
INHERITS_FROM	: PRODUCT_ID_MODEL

NAME : LATITUDE
 CLASS : DATA_FIELD
 DEFINITION : ‘It represents the latitude used for the centre coordinate’.
 INHERITS_FROM : LATITUDE_MODEL

NAME : LONGITUDE
 CLASS : DATA_FIELD
 DEFINITION : ‘It represents the longitude used for the centre coordinate’.
 INHERITS_FROM : LONGITUDE_MODEL

NAME : W_IMAGE_SIZE
 CLASS : CONSTANT
 DEFINITION : ‘It represents the number of pixels for an image taken from spacecraft W2’.
 SHORT_DEFINITION : ‘Spacecraft W2 Image pixel’
 RELATION : (‘size of’, DATA_2)
 DATA_TYPE : INTEGER
 CONSTANT_VALUE : 1 440 000

NAME	: DATA_2
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents an image taken from spacecraft W2’.
SHORT_DEFINITION	: ‘Spacecraft W2 Image’
COMMENT	: ‘The image is an array of W_IMAGE_SIZE items called DATA_2_PIXEL’
COMPONENT	: DATA_2_PIXEL (1 .. W_IMAGE_SIZE)
KEYWORD	: ‘IMAGE’
DATA_TYPE	: COMPOSITE
NAME	: DATA_2_PIXEL
CLASS	: DATA_FIELD
DEFINITION	: ‘It represents a pixel belonging to the image taken from spacecraft W2’.
SHORT_DEFINITION	: ‘Spacecraft W2 Image pixel’
DATA_TYPE	: INTEGER
RANGE	: (0 , 255)

ANNEX B**MAPPING OF THE CONCEPTS BETWEEN THIS
RECOMMENDATION AND THE ISO/IEC 11179-3:1994 STANDARD**(This annex **is not** part of the Recommendation)**1) Mapping between the descriptors**

DEDSL	ISO 11179-3:1994
Attribute_name	Name
Attribute_definition	Definition
Attribute_obligation	
Attribute_maximum_occurrence	Maximum occurrence
Attribute_condition	Condition
Attribute_value_type	Data type
	Maximum size
	Character set
	Language
Attribute_comment	Comment
Attribute_value_example	
Attribute_default_value	
Attribute_inheritance	
Attribute_enumeration_values	
Attribute_maximum_size	
Attribute_scope	

2) Mapping between the attributes

DEDSL	ISO 11179-3:1994
Name	Name
	Identifier
	Version
	Registration Authority
Alias	Synonymous name
Alias	Context
Class	
Definition	Definition
Short_definition	
Comment	Comments
	Classification scheme—Class name
Units	
Specific_instance	
Inherits_from	
Component	
Keyword	Keyword
Relation	Related data reference
Relation	Type of relationship
	Representation class
Language	Form of representation

CCSDS RECOMMENDATION FOR DEDSL—ABSTRACT SYNTAX

DEDSL	ISO 11179-3:1994
Case_sensitivity	Form of representation
Data_type	Data type of data element values
	Maximum size of data element values
	Minimum size of data element values
	Layout of representation
Data_type	Code set
Enumeration_values	
Enumeration_meaning	
Enumeration_convention	
Range	
Text_size	
Constant_value	
	Responsible organisation
	Registration status
	Submitting organisation
Dictionary_name	
Dictionary_definition	
External_dictionary_reference	
Text_field_character_set	
Dictionary_version	
Dictionary_identifier	
Dedsl_version	

ANNEX C

INFORMATIVE REFERENCES

(This annex **is not** part of the Recommendation)

This annex provides a list of references that may be valuable to the user of this Recommendation as background material or to provide implementation guidelines for using this Recommendation.

- [C1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-7. Yellow Book. Issue 7. Washington, D.C.: CCSDS, November 1996.
- [C2] *The Data Description Language EAST—Specification (CCSD0010)*. Recommendation for Space Data System Standards, CCSDS 644.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1997. (FDIS 15889)
- [C-3] *The Data Description Language EAST—A Tutorial*. Report Concerning Space Data System Standards, CCSDS 645.0-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, May 1997.
- [C-4] *Standard Formatted Data Units—A Tutorial*. Report Concerning Space Data System Standards, CCSDS 621.0-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, May 1992.
- [C-5] *Standard Formatted Data Units—Control Authority Procedures Tutorial*. Report Concerning Space Data System Standards, CCSDS 631.0-G-2. Green Book. Issue 2. Washington, D.C.: CCSDS, November 1994.
- [C-6] *Hierarchical Data Format (HDF)*. Version 4.0r. National Centre for Supercomputing Applications (NCSA).
- [C-7] *Common Data Format (CDF)*. Version 2.5.19a. National Space Science Data Center, May 17, 1996.
- [C-8] *Time Code Formats*. Recommendation for Space Data System Standards, CCSDS 301.0-B-2. Blue Book. Issue 2. CCSDS April 1990. (ISO 11104)
- [C-9] *UNIDATA Units Package*. NCAR, Version 1.11.5, August 18, 1997.
- [C-10] *Information Processing—Representation of SI and other units in systems with limited character sets*. ISO 2955-1983. Geneva: ISO, 1983.
- [C-11] *Information Technology—Specification and standardisation of data elements—part 3: Basic attributes of data elements*. ISO/IEC 11179-3:1994. Geneva: ISO, 1994.