



CCSDS

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

**PRODUCER-ARCHIVE
INTERFACE
SPECIFICATION (PAIS)**

RECOMMENDED STANDARD

CCSDS 651.1-B-1

BLUE BOOK

February 2014

Recommendation for Space Data System Standards

**PRODUCER-ARCHIVE
INTERFACE
SPECIFICATION (PAIS)**

RECOMMENDED STANDARD

CCSDS 651.1-B-1

BLUE BOOK
February 2014

AUTHORITY

Issue:	Recommended Standard, Issue 1
Date:	February 2014
Location:	Washington, DC, USA

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 documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-3), 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
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. 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 **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members 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 **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than three years from its date of issuance, this **Recommended Standard** 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 **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

This Recommended Standard is a technical Recommendation providing the abstract syntax and an XML implementation of descriptions of data to be sent to an archive. These descriptions are negotiated agreements between the data Producer and the Archive that facilitate production of agreed data by the Producer and validation of received data by the Archive. This Recommended Standard includes an abstract syntax for describing how these data will be aggregated into packages for transmission and one concrete implementation for the packages based on the XML Formatted Data Unit (XFDU) standard (see reference [1]). This will fulfill parts of the process defined in the *Producer Archive Ingest Methodology Abstract Standard (PAIMAS)* (see reference [2]).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-3). Current versions of CCSDS documents are maintained at the CCSDS Web site:

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

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

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

Member Agencies

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

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 651.1-B-1	Producer-Archive Interface Specification (PAIS), Recommended Standard, Issue 1	February 2014	Original issue

CONTENTS

<u>Section</u>	<u>Page</u>
1 INTRODUCTION.....	1-1
1.1 PURPOSE AND SCOPE.....	1-1
1.2 APPLICABILITY.....	1-1
1.3 RATIONALE.....	1-2
1.4 CONFORMANCE.....	1-2
1.5 DOCUMENT STRUCTURE	1-2
1.6 DEFINITIONS.....	1-4
1.7 NOMENCLATURE	1-7
1.8 REFERENCES	1-8
2 OVERVIEW	2-1
2.1 GENERAL FRAMEWORK.....	2-1
2.2 FORMALLY DESCRIBING DATA OBJECTS FOR TRANSFER	2-2
2.3 SIP CREATION AND VALIDATION	2-4
3 DESCRIPTIONS OF PRODUCER DATA	3-1
3.1 GENERAL.....	3-1
3.2 TRANSFER OBJECT TYPE DESCRIPTOR SPECIFICATION (CCSD0014) ...	3-1
3.3 COLLECTION DESCRIPTOR SPECIFICATION (CCSD0015)	3-16
3.4 SPECIALIZATION OF THE DESCRIPTOR MODELS	3-23
4 DEFINITION OF CONSTRAINTS ON TYPES OF SIPS (CCSD0016).....	4-1
4.1 OVERVIEW	4-1
4.2 ABSTRACT SIP CONSTRAINTS	4-1
4.3 SIP SEQUENCING CONSTRAINTS IMPLEMENTATION.....	4-3
5 ABSTRACT SIP SPECIFICATION (CCSD0017)	5-1
5.1 OVERVIEW	5-1
5.2 SIP MODEL SPECIFICATION	5-1
5.3 DISCUSSION	5-8
5.4 SIP MODEL SPECIALIZATION	5-10
6 SIP IMPLEMENTATION	6-1
6.1 OVERVIEW	6-1
6.2 SIP IMPLEMENTATION USING XFDU	6-1
6.3 SPECIALIZATION OF THE SIP IMPLEMENTATION IN AN XFDU.....	6-8

CONTENTS (continued)

<u>Section</u>	<u>Page</u>
ANNEX A PAIS XML SCHEMAS (NORMATIVE)	A-1
ANNEX B LEGEND FOR XML FIGURES (INFORMATIVE)	B-1
ANNEX C INFORMATIVE REFERENCES (INFORMATIVE)	C-1
ANNEX D MANAGEMENT OF IDS (INFORMATIVE)	D-1
ANNEX E SECURITY, SANA, AND PATENT CONSIDERATIONS (INFORMATIVE)	E-1
ANNEX F SIP TO XFDU MAPPING EXAMPLE (INFORMATIVE)	F-1

Figure

2-1 PAIS General Process	2-2
2-2 An Example of the Entities and Their Relationships Involved in Creating the Formal Specifications	2-4
2-3 SIP Process	2-6
3-1 First Decomposition Level of ‘transferObjectTypeDescriptor’	3-3
3-2 Complete Decomposition of the ‘transferObjectTypeDescriptor’	3-4
3-3 Transfer Object Type Descriptor ‘identification’	3-6
3-4 Transfer Object Type Descriptor ‘description’	3-8
3-5 Transfer Object Type Descriptor ‘relation’	3-9
3-6 Transfer Object Type Descriptor ‘groupType’	3-15
3-7 Transfer Object Type Descriptor ‘any’	3-16
3-8 First Decomposition Level of ‘collectionDescriptor’	3-18
3-9 Complete Decomposition of a ‘collectionDescriptor’	3-18
3-10 Collection Descriptor ‘identification’	3-20
3-11 Collection Descriptor ‘description’	3-21
3-12 Collection Descriptor ‘relation’	3-22
3-13 Collection Descriptor ‘any’	3-22
3-14 Specialization of the Descriptor Models	3-23
4-1 Decomposition of ‘sipConstraints’	4-3
5-1 Abstract View of the SIP Container	5-2
5-2 Abstract View of the Transfer Object Container	5-4
5-3 Abstract View of the Transfer Object Group Container	5-5
5-4 Abstract View of the Data Object Container	5-7
5-5 Abstract View of SIP, Transfer Object, Transfer Object Group, and Data Object	5-9
5-6 Specialization of the SIP Model	5-10
6-1 SIP and XFDU Schemas Constrain the XFDU XML Manifest to Implement the SIP as an XFDU	6-2
6-2 SIP Global Information	6-3
6-3 SIP Transfer Object Identification and Status	6-4
6-4 SIP Transfer Object Group Identification	6-4
6-5 SIP Data Object Identification	6-5

CONTENTS (continued)

<u>Figure</u>	<u>Page</u>
6-6 SIP Transfer Object To Delete	6-5
6-7 Any Extension Type	6-5
6-8 sipGlobalInformation Mapped to XFDU.....	6-6
6-9 sipTransferObject Mapped to XFDU	6-7
6-10 Byte Stream Mapped to XFDU byteStream Element.....	6-8
B-1 Legend for XML Figures.....	B-2

1 INTRODUCTION

1.1 PURPOSE AND SCOPE

The purpose of this Recommended Standard is to provide a standard method for formally defining the digital information objects to be transferred by an information Producer to an Archive and for effectively packaging these objects in the form of **Submission Information Packages (SIPs)**. This supports effective transfer and validation of SIP data.

This Recommended Standard fits into the context defined by:

- The *Reference Model for an Open Archival Information System (OAIS)* Recommended Standard (see reference [3]).
- The *Producer-Archive Interface Methodology Abstract Standard (PAIMAS)* Recommended Standard (see reference [2]).
- The *XML Formatted Data Unit (XFDU) Structure and Construction Rules* Recommended Standard (see reference [1]).

The PAIMAS Recommended Standard (see reference [2]) defines a methodology based on the four following phases: Preliminary, Formal Definition, Transfer, Validation.

This Recommended Standard applies specifically to the implementation of the main part of the Formal Definition Phase and the Transfer Phase, taking into account part of the Validation Phase.

The proposed implementation should help in the automation and management of the Transfer and Validation Phases.

The proposed implementation may also be used, to some extent, for the Preliminary Phase.

This Recommended Standard does not exclude other PAIMAS implementation Recommended Standards.

1.2 APPLICABILITY

The implementation defined in this document applies to any **Producer-Archive Project**. It is specifically applicable to those organizations and individuals who create information that may need Long-Term Preservation and to organizations making information available for the Long Term.

This application is relevant only if both partners in the Producer-Archive Project agree with a shared implementation as defined in this document.

1.3 RATIONALE

This Recommended Standard aims at overcoming significant difficulties encountered during transactions between information Producers and the Archives.

Regarding the Formal Definition Phase, this Recommended Standard should enable:

- the Producer to share with the Archive a sufficiently precise, unambiguous definition of the different Digital Objects to be produced and transferred, including possibly the order in which they should be transferred;
- the Archive to ensure there is sufficient information to process the Digital Objects which will be received.

Regarding the Transfer Phase, this Recommended Standard should enable a precise definition of the SIPs to be exchanged.

Regarding the Validation Phase, this Recommended Standard should enable the use of tools for systematically validating that the Digital Objects received are those expected, and that they conform to the level of detail previously agreed.

1.4 CONFORMANCE

An xml implementation is considered ‘Description Conformant’ if it conforms to the corresponding semantics and implementation specifications defined in sections 3 and 4.

A concrete SIP implementation is considered ‘Abstract SIP Conformant’ if it conforms to the semantic specification of section 5.

A concrete SIP implementation is considered ‘XFDU PAIS SIP Conformant’ if it conforms to the semantic specification of section 5 and the implementation specification of section 6.

Concrete SIP implementations in formats other than XFDU are allowed. They will not be considered ‘XFDU PAIS SIP Conformant’, but they could be ‘Abstract SIP Conformant’. It is possible that additional concrete PAIS SIP implementations may be standardized in the future.

1.5 DOCUMENT STRUCTURE

1.5.1 HOW TO READ THIS DOCUMENT

All readers should study subsections 1.1 (Purpose and Scope), 1.2 (Applicability), and 1.4 (Conformance) in order to understand the objectives and applicability of this Recommended Standard.

Readers seeking an overview of the specification should also read section 2 (Overview).

Those who will implement the specification should read the entire document.

NOTE – A working knowledge of the concepts defined in the PAIMAS (reference [2]), and of the XFDU structure and construction rules (reference [1]) may be helpful in order to understand this Recommended Standard.

1.5.2 ORGANIZATION BY SECTION

Section 0 defines the purpose, scope, applicability, rationale and definitions for terminology used in this Recommended Standard. It also specifies what is required for conformance to this standard.

Section 2 contains a general overview of the specification. This overview describes the general framework of the PAIS using terms and concepts from reference [2]. It addresses a formal description of the Data Objects for transfer and the creation and validation of the SIPs containing those Data Objects.

Section 3 analyzes in detail the model of the Data Objects to be transferred by the Producer to the Archive. These Data Objects are organized into collections and Transfer Objects that are described in detail. Each description is divided into an abstract view, followed by the implementation view (partial schemas).

Section 4 describes the two different constraints that apply to the SIPs. The first one specifies the content authorized for each type of SIP. The second one specifies, if necessary, in which order the SIPs must be delivered. Each part is divided into an abstract view, followed by the implementation view (partial schemas).

Section 5 describes in detail the different SIP organizing entities, called containers.

Section 6 specifies a concrete SIP implementation using the XFDU (reference [1], partial schemas), and explains how it maps to the XFDU XML schema.

The annexes listed here are normative or informative:

- Annex A contains the full XML schemas for this specification and is normative. XML schema diagrams are presented throughout other sections of the book. If an XML schema diagram differs from the schema, the schema is considered to be the ruling entity.
- Annex B is a legend for symbols for the XML Authority Diagrams that appears in sections 3 to 6 of this document and is informative.
- Annex C contains the informative references.
- Annex D provides a table showing the management of the different identifiers defined in this document and is informative.

- Annex E discusses Security, Space Assigned Numbers Authority (SANA), and Patent Considerations and is informative.
- Annex F provides an example SIP, with Descriptors, and the mapping of the SIP to the XFDU, and is informative.

1.6 DEFINITIONS

1.6.1 ACRONYMS AND ABBREVIATIONS

This subsection defines the acronyms and abbreviations which are used throughout this Recommended Standard:

AIP	Archival Information Package
CCSDS	Consultative Committee for Space Data Systems
CCSDS CA	CCSDS Control Authority
CRC	Cyclic Redundancy Check
DED	Data Entity Dictionary
DTD	Document Type Definition
FTP	File Transfer Protocol
GB	Gigabyte
ID	Identifier
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
KB	Kilobyte
MB	Megabyte
MIME	Multipurpose Internet Mail Extensions
MOT	Model of Objects for Transfer
OAIS	Open Archival Information System
PAIMAS	Producer Archive Interface Methodology Abstract Standard
PAIS	Producer-Archive Interface Specification
PB	Petabyte
SIP	Submission Information Package
TB	Terabyte
TOTD	Transfer Object Type Descriptor
URL	Universal Resource Locator
XFDU	XML Formatted Data Unit
XML	eXtensible Markup Language

1.6.2 GLOSSARY OF TERMS

OAIS terminology as defined in references [2] is used throughout this recommendation. Following is a short glossary of the OAIS terminology indispensable for this document. The terminology used is fully defined in references [2], [3], and [C3], except for the definitions printed in italics which are defined in this document. Only brief definitions are provided here. This terminology does not seek to replace existing terminology in the various domains related to archiving. Each domain should be able to apply this methodology while retaining their specific terminology. When first used in the following sections, the terms defined in the terminology are shown in bold.

Moreover, it is assumed that it is not necessary for the Producer to know and understand the information model and the typology of the OAIS information categories in detail, such as Content Information, Representation Information, Preservation Description Information, etc. Indeed, it is the Archive's task to create AIPs from the SIPs transferred and thus to establish the suitable link between a given object coming from the Producer and any particular information category in the AIP within which this object will be inserted. To establish a dialog, the Producer and the Archive must agree on a common terminology and a common understanding of the associated concepts.

Archive: An organization that intends to preserve information for access and use by a Designated Community.

Collection Descriptor: A set of attributes that describes a view of a single collection of data and that identifies the parent collection of which it is a part.

Data Object: Either a Physical Object or a Digital Object.

Data Object Type: A set of characteristics describing a Data Object (such as the size of this object and the description of its content). Typically there will be multiple Data Objects conforming to the same Data Object Type.

Descriptor: Either a Collection Descriptor or a Transfer Object Type Descriptor.

Descriptor Model: A model that defines the mandatory and optional attributes needed for a Collection Descriptor or a Transfer Object Type Descriptor.

Digital Object: An object composed of a set of bit sequences.

Fixity Information: The information which documents the mechanisms that ensure that the Content Information Object has not been altered in an undocumented manner. An example is a Cyclical Redundancy Check (CRC) code for a file.

Information: Any type of knowledge that can be exchanged. In an exchange, it is represented by data. An example is a string of bits (the data) accompanied by a description of how to interpret a string of bits as numbers representing temperature observations measured in degrees Celsius (the Representation Information).

Information Package: A conceptual container composed of optional Content Information and optional associated Preservation Description Information. Associated with this Information Package is Packaging Information used to delimit and identify the Content Information and Package Description Information used to facilitate searches for the Content Information.

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 (see reference [C3]).

Model of Objects for Transfer (MOT): *The set of all Descriptors for a given Producer-Archive Project. It is used jointly by the Producer and the Archive to provide a common and understandable hierarchical view of the Producer's Data Objects to be transferred and their organization into collections, and it supports possible additional relationships among them. The hierarchy may be viewed as a tree having leaf and non-leaf nodes. The Data Objects to be transferred, organized as 'Transfer Objects,' are represented by the leaves of the MOT. Thus the nodes of the MOT have a different meaning depending on whether they are leaves or not:*

- *A leaf node corresponds to a single Transfer Object Type and therefore one exists for each Transfer Object Type Descriptor.*
- *A non-leaf node corresponds to a collection view of Transfer Object Types, or of a collection of collections. A non-leaf node exists for each Collection Descriptor.*

Producer: The role played by those persons or client systems who provide the information to be preserved. This can include other OAISes or internal OAIS persons or systems.

Producer-Archive Project: A Producer-Archive Project is a set of activities and the means used by the information Producer as well as the Archive to ingest a given set of information into the Archive.

Submission Agreement: The agreement reached between an OAIS and the Producer that specifies a data model, and any other arrangements needed, for the Data Submission Session. This data model identifies format/contents and the logical constructs used by the Producer and how they are represented on each media delivery or in a telecommunication session.

Submission Information Package (SIP): An Information Package that is delivered by the Producer to the OAIS for use in the construction or update of one or more AIPs and/or the associated Descriptive Information.

Transfer Object: *A set of one or more Transfer Object Groups containing at least one Data Object that are to be transferred to the Archive.*

Transfer Object Group: *A set of zero or more Data Objects and zero or more Transfer Object Groups.*

Transfer Object Group Type: A set of characteristics describing a Transfer Object Group. Typically there can be multiple Transfer Object Groups conforming to the same Transfer Object Group Type.

Transfer Object Type: A set of characteristics describing a Transfer Object (such as the size of this object, the description of its content, and its makeup in terms of one or more Data Object Types). Typically there can be multiple Transfer Objects conforming to the same Transfer Object Type.

Transfer Object Type Descriptor: A set of attributes that describes a Transfer Object Type and that identifies the parent collection of which it is a part.

1.7 NOMENCLATURE

1.7.1 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Standard:

- a) the words 'shall' and 'must' imply a 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.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.7.2 INFORMATIVE TEXT

In the normative sections of this document (sections 3-6), informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

- Overview;
- Background;
- Rationale;
- Discussion.

1.8 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommended Standard. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommended Standards.

- [1] *XML Formatted Data Unit (XFDU) Structure and Construction Rules*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 661.0-B-1. Washington, D.C.: CCSDS, September 2008. [Equivalent to ISO 13527:2010.]
- [2] *Producer-Archive Interface Methodology Abstract Standard*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 651.0-M-1. Washington, D.C.: CCSDS, May 2004. [Equivalent to ISO 20652:2006.]
- [3] *Reference Model for an Open Archival Information System (OAIS)*. Issue 2. Recommendation for Space Data System Practices (Magenta Book), CCSDS 650.0-M-2. Washington, D.C.: CCSDS, June 2012. [Equivalent to ISO 14721:2012.]

2 OVERVIEW

2.1 GENERAL FRAMEWORK

The general context for this Recommended Standard is that of the transfer of **Digital Objects** from a data **Producer** to an **Archive**. A methodology for specifying, performing, and validating this transfer is defined in the PAIMAS Recommended Standard (see reference [2]).

A key objective of the PAIS Recommended Standard is to provide a method to formally define the Digital Objects, along with their important inter-relationships, that are to be transferred by a data Producer to an Archive. This is performed during the Formal Definition Phase as defined in PAIMAS.

Another objective is to support the effective transfer of these objects in the form of **Submission Information Packages (SIPs)** as modeled in the OAIS Reference Model (see reference [3]). If these objectives are met, use of the PAIS Recommended Standard should facilitate validation by the Archive that all the objects expected have been received and that they conform to the characteristics expected. This is performed during the Transfer Phase and the Validation Phase.

A high-level view of the process involving use of this Recommended Standard is given in figure 2-1. This process shows a breakdown into two phases: Formal Definition and Transfer with Validation. **Data Objects** are formally defined using **Descriptors**, and then constraints on their transfer in terms of grouping and sequencing are addressed. This is negotiated with the Producer and results in the **Submission Agreement**. During the Transfer Phase the Data Objects are packaged into SIPs with links to their associated Descriptors, thus allowing the Archive to perform validation on each Data Object.

The PAIS Recommended Standard specifies how to construct Descriptors, how to define types of SIPs, and how to define any sequencing constraints among types of SIPs. It also specifies an abstract SIP and how to instantiate this within a standardized data package known as an XFDU (see reference [1]). It does not address the actual transfer of a SIP nor how the Archive does validation upon the received SIP. The extent of such validation will depend, in part, on the details of Descriptor implementations and the level of validation required by the **Producer-Archive Project**.

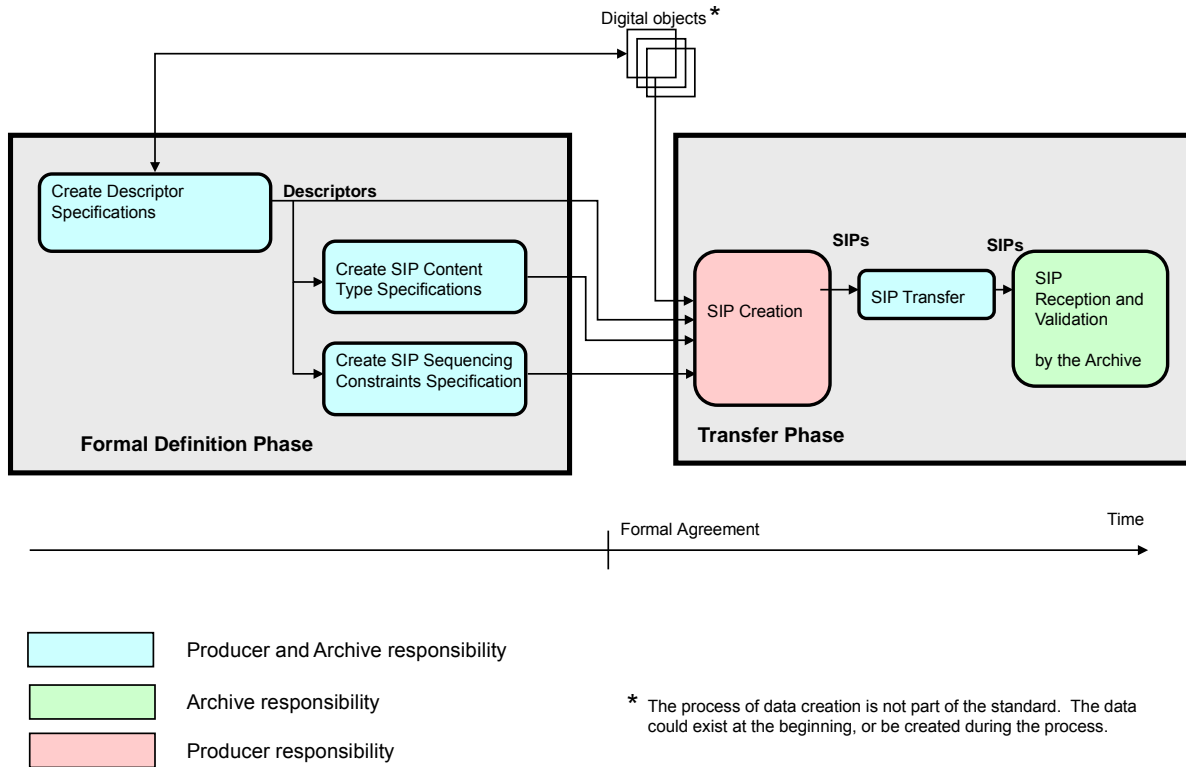


Figure 2-1: PAIS General Process

2.2 FORMALLY DESCRIBING DATA OBJECTS FOR TRANSFER

An example of the process for generating formal specification material is shown in figure 2-2.

One of the objectives of the Formal Definition Phase is to define the Data Objects, where hereafter a Data Object is understood to be one or more data files, to be transferred as possibly more complex data structures called **Transfer Objects**. Each Transfer Object is an instance of a particular **Transfer Object Type**. Each Transfer Object Type is described in detail by a **Transfer Object Type Descriptor**. A Transfer Object Type Descriptor consists of a set of mandatory, optional, and possibly user-defined attributes, and is generated for each type of Transfer Object. A collection of Transfer Object Types, or a collection of other collections, is described by a **Collection Descriptor**. It also consists of a set of mandatory, optional, and possibly user defined attributes. The Transfer Object Type and Collection Descriptors are expected to be complete enough to provide the information needed for an adequate description of different types of objects to be transferred to the Archive, and they are expected to be flexible enough to adapt to the specializations required by each Producer-Archive Project. One Collection Descriptor will be generated for each collection view. Both Transfer Object Type Descriptors and Collection Descriptors have attributes giving relationships among the Data Objects they are describing. The set of Descriptors for a Producer-Archive Project constitutes a **Model** called the **Model of Objects for Transfer (MOT)** for that project. Each Descriptor has a mandatory relationship-attribute, called 'parentCollection', that identifies the Collection Descriptor that provides a collection view of

the subject Descriptor's Data Objects. The resulting hierarchical view of Descriptors and the data they describe constitute an important aspect of the MOT.

It should be noted that the MOT is not an information organization model within the Archive. It is simply a way to enable the Producer and Archive to agree on the content of the information to be transferred. As such the MOT is the basis for negotiating a Submission Agreement. It should be noted also that each Descriptor has optional and possibly user defined attributes. This allows an Archive to tailor the two **Descriptor Models** (Transfer Object Type and collection) given in this Recommended Standard to meet the needs of describing the data to be transferred in a given Producer-Archive Project.

In the remainder of this document, the term 'Descriptor' is used for a Transfer Object Type Descriptor or a Collection Descriptor, and the phrase 'Transfer Object' is used to refer to an instance, while 'Transfer Object Type' or 'type of Data Object' is used to refer to the class of a Transfer Object or Data Object.

Section 3 contains the requirements from the Producer and the Archive to specify and implement the MOT during the Formal Definition Phase, described just above.

Another objective of the Formal Definition Phase is the consideration of whether some SIPs will be allowed to carry only some Transfer Object Types and not others, thereby defining SIP Content Type constraints, and whether some SIPs will need to be processed by the Archive before other SIPs, thereby defining SIP sequencing constraints. This is shown schematically in figure 2-2. This Recommended Standard requires that a Transfer Object, and therefore also a Data Object, shall not be divided between two or more SIPs. This is to facilitate management of the Transfer Object exchange.

Section 4 contains the requirements from the Producer and the Archive to specify and implement the SIP Content Type constraints and the sequencing constraints during the Formal Definition Phase.

At this stage of negotiation between the Producer and the Archive, some characteristics of the objects to be transferred may not be known. How and when these unknowns will be defined should be detailed in the Submission Agreement. However, both the Archive and the Producer should each have the necessary information to negotiate and approve the MOT and the noted constraints.

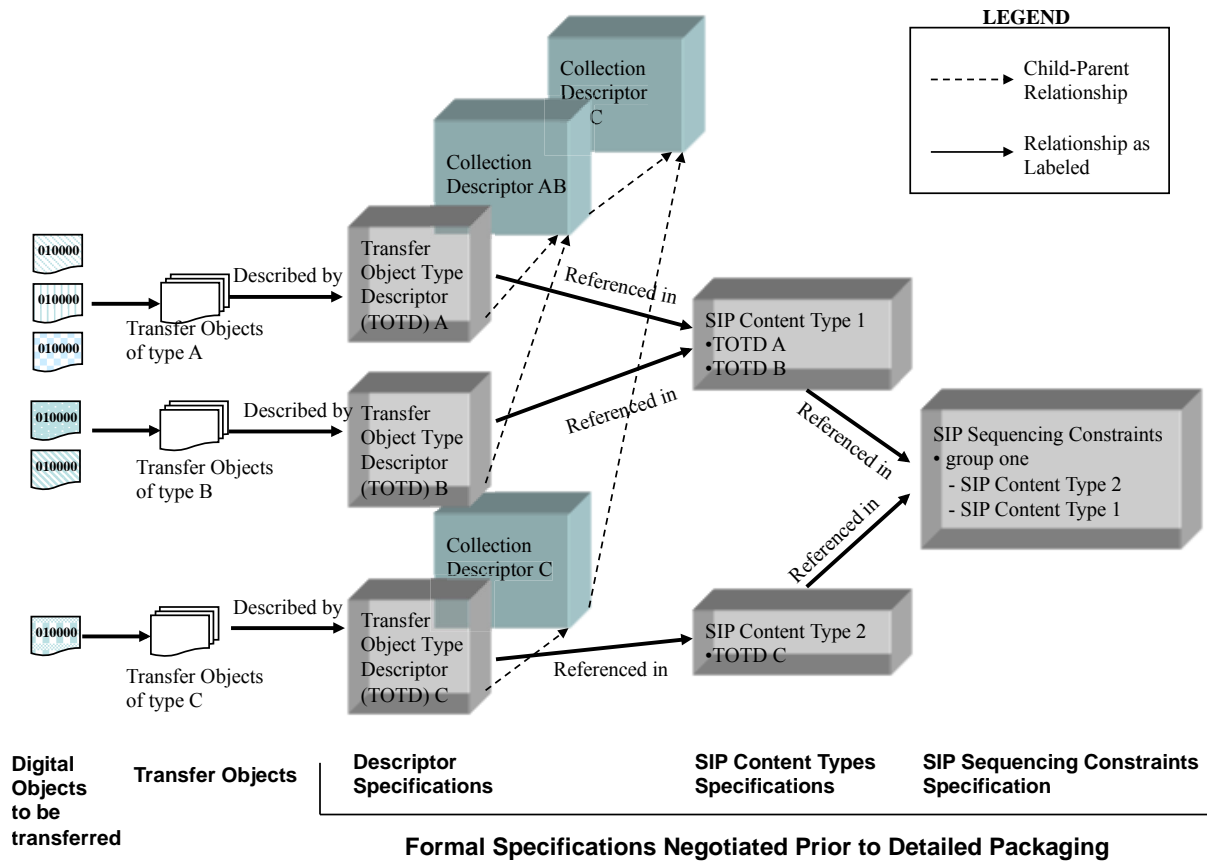


Figure 2-2: An Example of the Entities and Their Relationships Involved in Creating the Formal Specifications

2.3 SIP CREATION AND VALIDATION

Figure 2-3 shows the process of building the SIPs.

A SIP can simply be a coherent group of Data Objects which is transferred in the same package. A SIP may also contain a pointer to the Data Objects location in case the Data Objects are not physically transferred in the SIP. For example, a URL or FTP site may be referenced in the SIP.

These Data Objects may be grouped together in a SIP for very different reasons. For example:

- Because of Archive procedures, it may be necessary for the Archive to have several Data Objects made available within the same SIP package in order to be able to create the corresponding AIPs.
- It might be useful to optimize the transfer: if the Data Objects are very small, a decision may be taken to group several of these Data Objects together in a single package so as not to have too many packages.

Each Data Object is considered as a whole, which means in particular that the different bit sequences that make up the Data Object may not be separated into elements transferred in separate SIPs.

Introduction to SIP abstract model and mapping to XFDU:

Once the formal specifications have been negotiated and agreed, the next step is the creation of SIPs. Section 5 contains a complete description of the abstract SIP Model and section 6 contains its implementation with a mapping to XFDU.

Typically, the SIP carries one or more Transfer Objects and for each such object, identifies its Descriptor ID, the Data Objects that make up this Transfer Object, and the **Data Object Type** IDs for each of these Data Objects. The location of the Data Objects within the SIP must also be given. Since a Transfer Object may be as complex as a directory structure, a given Data Object may also need to be associated with a directory path and directory name. A Transfer Object may also carry an indication as to whether this is the last Transfer Object of this type that the Archive should receive. A SIP must also carry some global information that allows the Archive to recognize its origin and type.

During the Transfer and Validation Phases, the following scenario is expected to take place:

1. SIP Creation by the Producer.

During the Transfer Phase, the SIPs will be created from an Abstract SIP Model, for the objects to be delivered, according to the SIP Content Types defined during the Formal Definition Phase. This document specifies a SIP Abstract Model in section 5 and a concrete specification in section 6.

2. SIP Transfer between the Producer and the Archive.

This transfer is performed according to the agreements between the Producer and the Archive. It follows the SIP Sequencing Constraints to transfer the SIPs in the order specified. The SIP Sequencing Constraints have been defined during the Formal Definition Phase.

3. SIP Reception and Validation by the Archive.

After the reception of a SIP, the Archive performs the initial validations on the received SIP and may check that:

- the SIP conforms to the associated SIP Content Type;
- any Digital Object transferred is an expected Digital Object in the previously defined MOT;
- this Digital Object complies with the characteristics defined in the MOT, including the number expected, and the required structure of the data is preserved, such as a hierarchical structure of files in directories;

- the Digital Objects have been transferred in the right order (sequencing constraints);
- the Digital Objects conform to any supplied checksums.

Should any validations fail, the Archive follows previously agreed procedures for Archive-Producer communications.

This document does not specify how to define and perform the validation plans, nor the Producer-Archive communications.

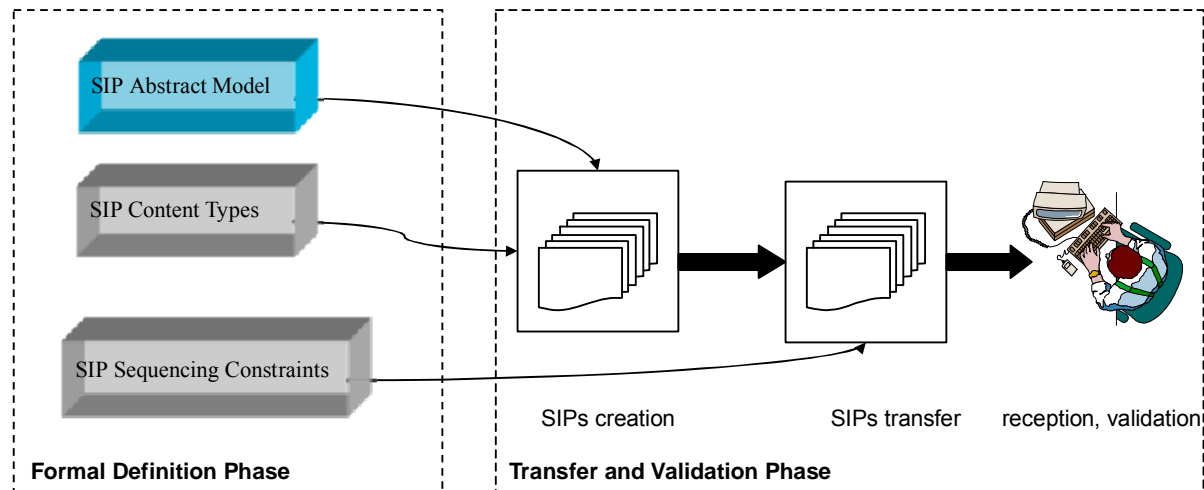


Figure 2-3: SIP Process

3 DESCRIPTIONS OF PRODUCER DATA

3.1 GENERAL

3.1.1 The following Descriptor Models shall be used for the modeling of data to be provided to an Archive:

- Model for the Transfer Object Type Descriptor (3.2);
- Model for the Collection Descriptor (3.3).

3.1.2 Each model shall be implemented using XML and shall conform to the XML schema given in annex A.

3.1.3 When XML schema diagrams are given, they are informative, as the XML schema of annex A is ruling should any conflict arise.

NOTES

1 Each model is specified using a language composed of attributes. The attributes are grouped into categories for ease of understanding. The meanings associated with the categories and the individual attributes are specified under an ‘abstract’ heading. This is immediately followed by a view of the concrete specification of the attributes using XML schema diagrams. The complete XML schema code is given in annex A and is normative.

2 Annex B gives the rules that apply for the notations of the XML schema diagrams.

3.1.4 The specifications that follow shall be understood in the context of a single Producer-Archive Project.

3.1.5 Each Producer-Archive Project shall be assigned a Producer-Archive Project identifier that must be unique across all such projects within the Archive.

NOTE – Annex D contains an overall table of identifiers management.

3.2 TRANSFER OBJECT TYPE DESCRIPTOR SPECIFICATION (CCSD0014)

3.2.1 TOP-LEVEL SPECIFICATION

3.2.1.1 Abstract View Top-Level Specification

3.2.1.1.1 The Transfer Object Type Descriptor shall be composed of a set of attributes and values that are assigned to these attributes in order to characterize the described Digital Object Types.

3.2.1.1.2 These attributes shall be grouped into the following top-level categories with the following definitions:

- **Descriptor Identification:** set of attributes supporting the unique identification of the Transfer Object Type Descriptor within the Producer-Archive Project. Users may incorporate additional attributes as needed to further specialize the Descriptor Identification.
- **Transfer Object Type Description:** set of attributes giving a high-level description of the Transfer Object Type, such as the title, the content, and the size. Users may incorporate additional attributes as needed to further specialize the Transfer Object Type Description.
- **Transfer Object Type Relationships:** set of attributes that identify the collection to which the Transfer Object Types under this Descriptor are a part, and optionally that specify directional relationships between these Transfer Object Types and other Data Object Types, Transfer Object Types, Transfer Object Group Types, and collections. These relationships are specified using identifiers as given within the Descriptors of the MOT. Users may incorporate additional attributes as needed to further specialize the Transfer Object Type Relationships.
- **Transfer Object Group Types:** set of attributes that identify one or several Group Types. A Group Type is a set of characteristics describing zero or more Data Object Types, even complex Data Objects Types made up of several parts. The Transfer Object Group Type may contain additional Transfer Object Group Types in support of complex structures such as directory structures. Users may incorporate additional attributes as needed, such as relative pathnames, to further specialize the Transfer Object Group Types.
- **User Defined Attributes:** new attributes that users may incorporate as needed to further specialize the Transfer Object Type Descriptor.

NOTES

- 1 In the following, each attribute has a mandatory or optional status, and has an occurrence. In the text, the occurrence is indicated between parentheses (min..max) after the name of the attribute; a zero value means that the attribute is optional, and it is mandatory otherwise.
- 2 The attributes themselves belong to groups that can be mandatory or optional. Inside an optional group, an attribute may be mandatory meaning that the attribute must be present if there is an instance of that group. (On the contrary, inside a mandatory group an attribute may be optional.)

3.2.1.2 XML Schema of Top-Level Specification

3.2.1.2.1 The Transfer Object Type Descriptor shall conform to the complete XML schema given in A2, and common types given in A1.

3.2.1.2.2 The Transfer Object Type Descriptor XML schema shall contain XML elements corresponding to the top-level categories as specified in the abstract view given above.

3.2.1.3 Discussion

Figure 3-1 provides the schema diagram for this first-level decomposition of the Transfer Object Type Descriptor. Figure 3-2 provides the schema diagram for the complete decomposition of the Transfer Object Type Descriptor. Descriptor attributes are implemented as XML elements.

In the next section the semantics for each attribute in a category are given along with a view of the subset of the XML schema that applies.

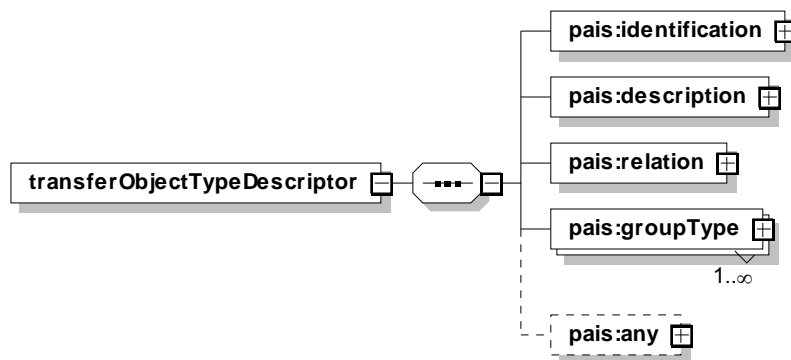


Figure 3-1: First Decomposition Level of ‘transferObjectTypeDescriptor’

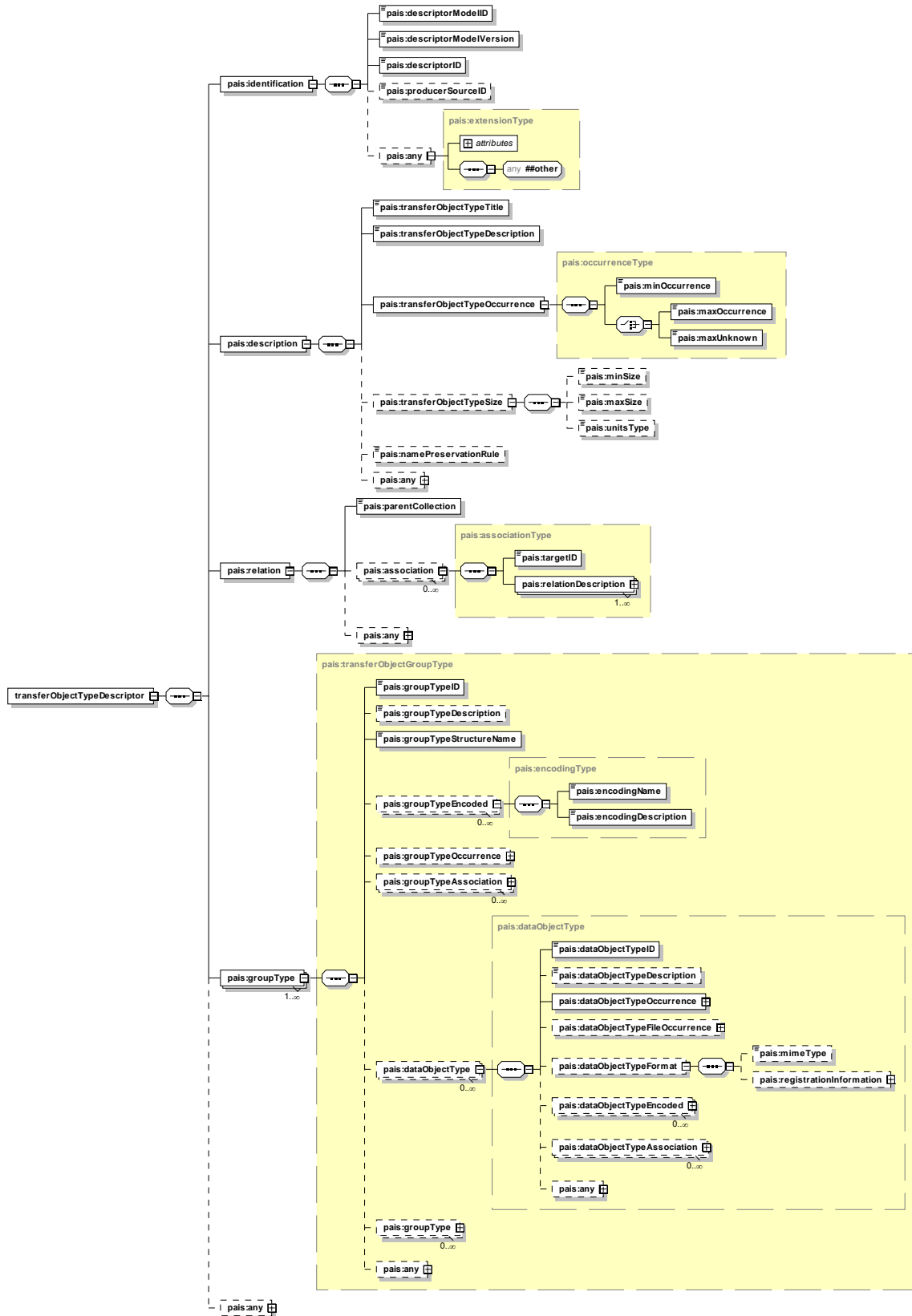


Figure 3-2: Complete Decomposition of the ‘transferObjectTypeDescriptor’

3.2.2 DETAILED DESCRIPTION

3.2.2.1 General

3.2.2.1.1 Each attribute in the identified categories shall have the semantics as specified in the following subsections having the introductory phrase ‘Abstract view’.

3.2.2.1.2 Each attribute in the identified categories shall be implemented in accordance with the XML schemas as identified in the following subsections having the introductory phrase ‘Implementation: XML schema’.

3.2.2.2 Transfer Object Type Identification

– **Abstract View:**

Identification (1..1)

- **Descriptor Model ID (1..1):** identifier of the Transfer Object Type Descriptor Model upon which this Descriptor is based. It may be the Descriptor Model as given in the standard or it may be a specialized version as described in detail in 3.4. It shall be unique across the Producer-Archive Project and shall be provided by the Archive.

The standard value for this ID, when no specialization has been employed, is ‘CCSD0014’. When specialization has been employed, the value for the Descriptor Model ID should be agreed between the Producer and the Archive.

- **Descriptor Model Version (1..1):** identifier of the version of the Descriptor Model. This allows tracking updates to the identified Descriptor Model.

The standard value for this ID, when the Descriptor Model ID is ‘CCSD0014’ and no revisions to this standard model have occurred, is ‘V1.0’. When specialization has been employed, the value for the Descriptor Model Version shall be agreed between the Producer and the Archive.

- **Descriptor ID (1..1):** identifier of this Descriptor within the Producer-Archive Project. Uniqueness shall be checked by the Archive. This identifier may be used by a relationship attribute to refer to this Transfer Object Type.
- **Producer Source ID (0..N):** identifier of the Producer Source sufficient to enable the Archive to identify the origin of the SIP and to send an acknowledgement. There may be multiple Producer Sources sending SIPs to the Archive within a given Producer-Archive Project. The form of the Producer Source ID shall be agreed between Producer and Archive. If specified, only those Producer Sources identified here may include Transfer Objects, of this Transfer Object Type, in SIPs.
- **Any (0..N):** mechanism that allows any new attribute(s) to be added as needed.

– **Implementation: XML Schema**

The identification attributes shall be implemented as the ‘identification’ element as specified in the XML schema given in A2, and common types given in A1.

NOTE – Figure 3-3 is a schema diagram for the ‘identification’ element.

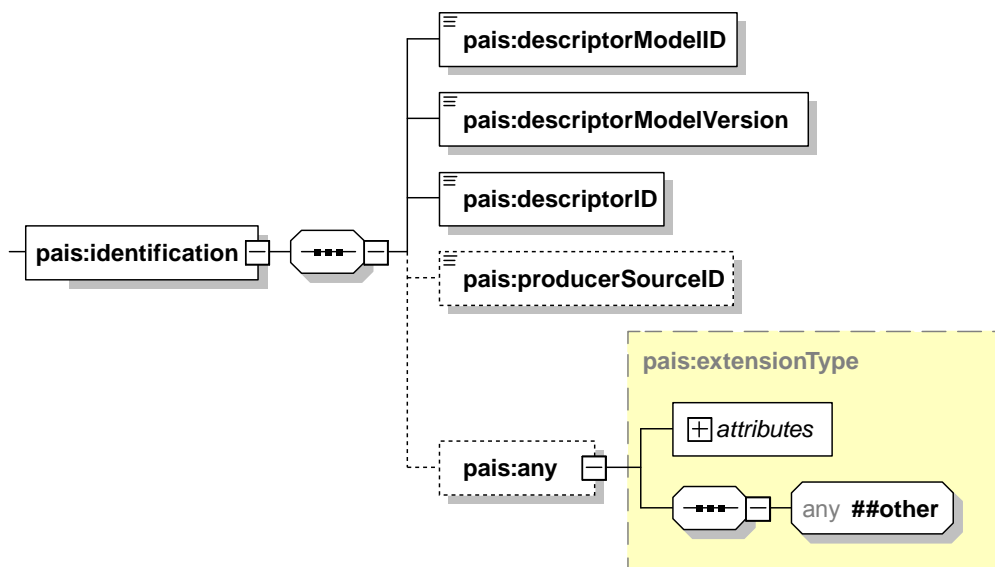


Figure 3-3: Transfer Object Type Descriptor ‘identification’

3.2.2.3 Transfer Object Type Description

– **Abstract View:**

Description (1..1)

- **Transfer Object Type Title (1..1):** extensive descriptive phrase used as the name of the Transfer Object Type.
- **Transfer Object Type Description (1..1):** explanatory text describing the meaning of the Transfer Object Type.
- **Transfer Object Type Occurrence (1..1):** number of instances (Transfer Objects) of this Transfer Object Type within the Producer-Archive Project. This may be expressed as one, or more, or as a range of values. This number may not be known at the time of descriptor creation. In the case of a unique value, the Minimal Occurrence and the Maximal Occurrence shall have the same value.
 - **Minimal Occurrence (1..1):** non-negative integer value (\leq maximal occurrence).

Only one of the two following attributes ‘Maximal Occurrence’ or ‘Maximal Unknown’ shall be used and is mandatory.

- **Maximal Occurrence (1..1):** non-negative integer value (\geq minimal occurrence).
 - **Maximal Unknown (1..1):** empty attribute when used.
 - **Transfer Object Type Size (0..1):** characterization of the size range for Transfer Objects of this type. The size range may not be known at the time of descriptor creation and therefore the corresponding attribute may be omitted.
 - **Minimal Size (0..1):** float value (\leq maximal size when specified).
 - **Maximal Size (0..1):** float value (\geq minimal size when specified).
 - **Units Type (1..1):** list of units: KB, MB, GB, TB, PB.
 - **Name Preservation Rule (0..1):** specification from the Producer describing how to recognize and/or create the preservable name to be given to each Data Object. It shall apply to all the Data Objects under this Transfer Object Type. It may assist the Producer in creating the SIPs and it may assist the Archive in preserving appropriate information.
 - **Any (0..N):** mechanism that allows any new attribute(s) to be added as needed.
- **Implementation: XML Schema**

The Description attributes shall be implemented as the ‘description’ element as specified in the XML schema given in A2, and common types given in A1.

NOTES

- 1 Figure 3-4 is a schema diagram for the ‘description’ element.
- 2 As shown in figure 3-4, the ‘transferObjectTypeOccurrence’ parent element is of type ‘occurrenceType’. (This type is used as well in the ‘groupType’ element.)

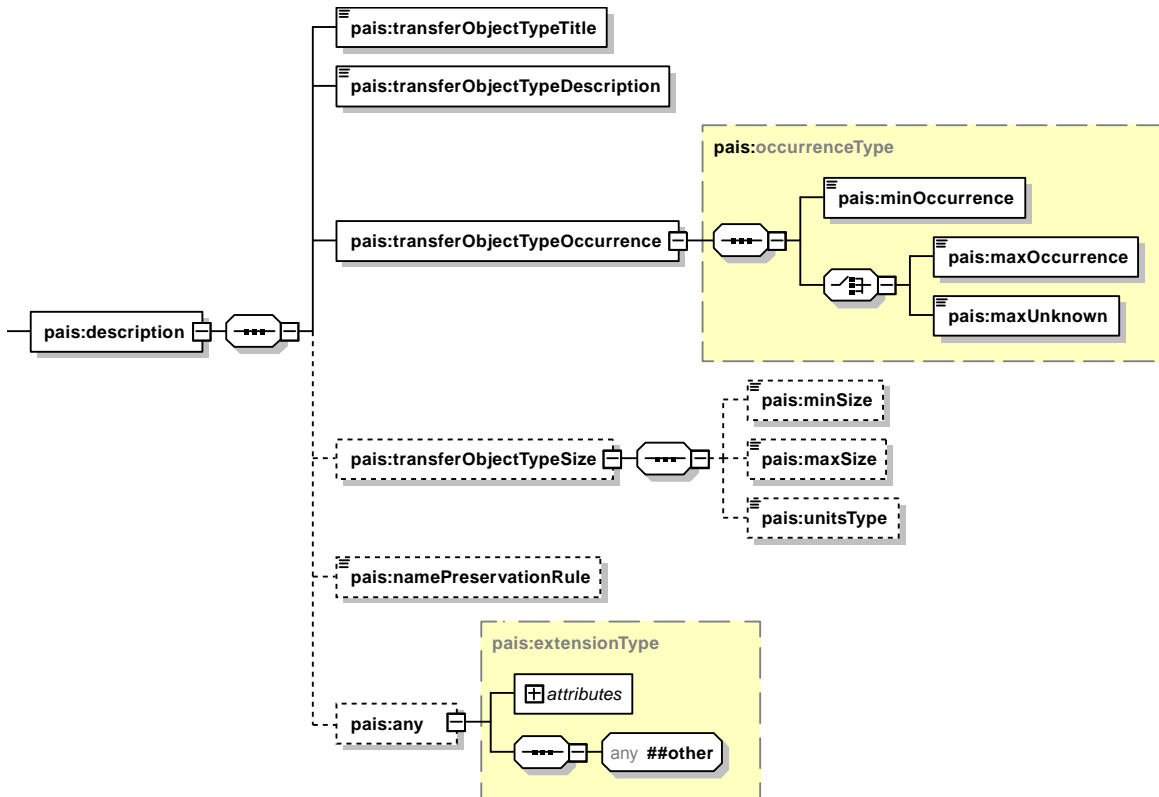


Figure 3-4: Transfer Object Type Descriptor ‘description’

3.2.2.4 Transfer Object Type Relationships

– Abstract View:

Relationships (1..1)

- **Parent Collection (1..1):** identifier of a Collection Descriptor that provides an aggregation view that includes this Transfer Object Type. The highest-level Descriptor in the MOT shall be a Collection Descriptor.
- **Association (0..N):** attribute used to describe a relationship from this Transfer Object Type under this Descriptor to the object type or collection having the identifier given by the Target ID attribute (targeted object types include Transfer Object Type, Transfer Object Group Type, and Data Object Type). These are transversal links specifying the type of relation.
 - **Target ID (1..1):** identifier of target object type or collection of the Producer-Archive Project to which the relationship is established. Possible target object types and their corresponding identifiers include: (Transfer Object Type, Descriptor ID), (Transfer Object Group Type, Transfer Object Group Type ID), (Data Object Type, Data Object Type ID), (collection, Descriptor ID).

- **Relation Description (1..N)**: description of the relationship or role played by this Transfer Object Type with respect to the target object type or collection.
 - **Relation Type (1..1)**: name for the relation (e.g., Data Entity Dictionary [DED], Syntax, Context, Provenance, Reference, Fixity Information) played by this object type or collection with respect to the target object type or collection.
 - **Relation Textual Description (0..1)**: text description of the relation.
- **Any (0..N)**: mechanism that allows any new attribute(s) to be added as needed.

– **Implementation: XML Schema**

The Relation attributes shall be implemented as the ‘relation’ element as specified in the XML schema given in A2, and common types given in A1.

NOTES

- 1 Figure 3-5 is a schema diagram for the ‘relation’ element.
- 2 As shown in figure 3-5, the ‘relation’ parent element is made up of the ‘association’ element of type ‘associationType’. This type is used as well in the ‘groupType’ element, ‘dataObjectType’ element, and in the relation element of the ‘collectionDescriptor’.

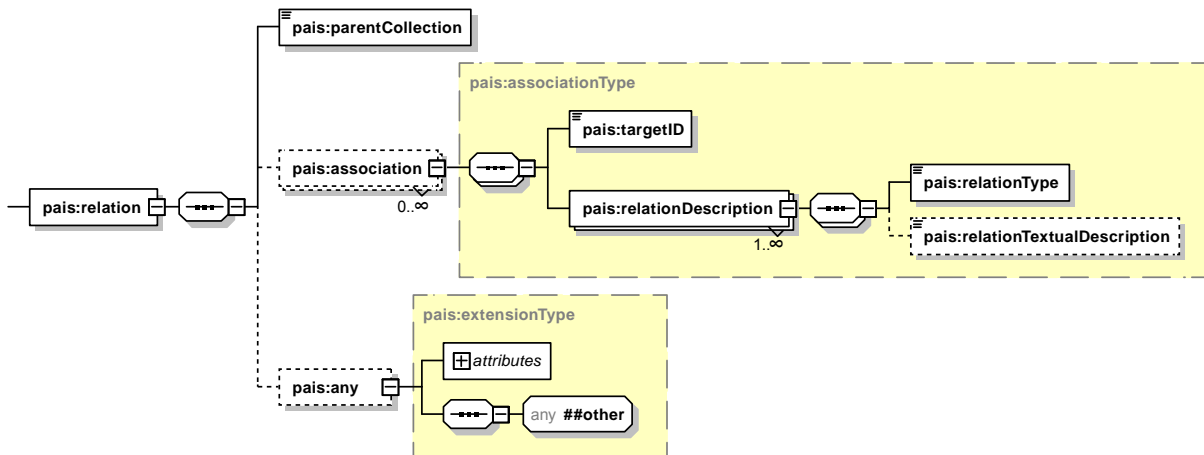


Figure 3-5: Transfer Object Type Descriptor ‘relation’

3.2.2.5 Transfer Object Group Types

– **Abstract View:**

Transfer Object Group Type (1..N)

- **Transfer Object Group Type ID (1..1)**: identifier of a Transfer Object Group Type that shall be used to distinguish this Group Type from other Group Types. It

shall be unique within a given Producer-Archive Project. Uniqueness shall be checked by the Archive. This identifier may be used by a relationship attribute to refer to this Transfer Object Group Type.

- **Transfer Object Group Type Description (0..1):** explanatory text describing the meaning of this Transfer Object Group Type.
- **Transfer Object Group Type Structure Name (1..1):** name for the type of Transfer Object Group Type, such as directory, set, sequence, undescribed, in the sense defined below, or other name not included in this list.

Directory: value that states that instances of groups and Data Objects at the same level within this Group Type are understood as being within a directory. A directory group shall have a name.

Set: value that states that instances of groups and Data Objects at the same level within this Group Type are understood to be without special organization among them. A set may or may not have a name.

Sequence: value that states that instances of groups and Data Objects at the same level within this Group Type are understood to have an order associated with them. Only Groups or only Data Objects are allowed to be in a sequence; they shall not be mixed. The way to recognize the order should be given in the associated Group Type Description. A sequence may or may not have a name.

Undescribed: value that states that the detailed modeling of this Group Type, and any data structure under this Group Type, has intentionally been left undescribed. Therefore this Group Type shall not include any Data Object Type specifications nor any other Group Type specifications. However, all other Group Type attributes may be present.

- **Transfer Object Group Type Encoded (0..N):** characterization of how groups of this type are encoded. It may be used iteratively to describe possible nested encodings, in which case they are described in the order of application. The result of encoding shall be a single Data Object. The possible presence of Data Object Type and any additional Transfer Object Group Type specifications under this Transfer Object Group Type specification shall be understood to comprise a detailed model of this group after the encoding has been reversed. With the use of this attribute, validation of the encoded content may not be possible because the resulting data structure will not carry any of the identifiers associated with the detailed modeling of this group. The following two attributes specify the details of the encoding:
 - **Encoding Name (1..1):** name for the type of encoding, such as zip, tar, or other name not included in this list.
 - **Encoding Description (1..1):** description or reference to the type of the encoding.

NOTE – For example a set of Data Objects viewed as a set and encoded in a tar file and then encoded in a zip file would be:

```
<groupTypeStructureName> set</groupTypeStructureName>
<groupTypeEncoded>
  <encodingName>tar</encodingName>
  <encodingDescription>application/x-tar</encodingDescription>
</groupTypeEncoded>
<groupTypeEncoded>
  <encodingName>gzip</encodingName>
  <encodingDescription>application/x-gzip</encodingDescription>
</groupTypeEncoded>
```

- **Transfer Object Group Type Occurrence (1..1):** number of instances (Transfer Object Groups) of this Transfer Object Group Type in each instance of its containing group or, if there is no containing group, in each containing Transfer Object Type instance. This may be expressed as one, or more, or as a range of values. This number may not be known at the time of descriptor creation. In the case of a unique value, the Minimal Occurrence and the Maximal Occurrence shall have the same value:
 - **Minimal Occurrence (1..1):** non-negative integer value (<=maximal occurrence).

Only one of the two following attributes ‘Maximal Occurrence’ or ‘Maximal Unknown’ shall be used and is mandatory:

 - **Maximal Occurrence (1..1):** non-negative integer value (>=minimal occurrence).
 - **Maximal Unknown (1..1):** empty attribute when used.
- **Transfer Object Group Type Association (0..N):** attribute used to describe a relationship from this Transfer Object Group Type to the object type or collection having the identifier given by the Target ID attribute (targeted object types include Transfer Object Type, Transfer Object Group Type, and Data Object Type). These are transversal links specifying the type of relation.
 - **Target ID (1..1):** identifier of target object type or collection of the Producer-Archive Project to which the relationship is established. Possible target object types and their corresponding identifiers include: (Transfer Object Type, Descriptor ID), (Transfer Object Group Type, Transfer Object Group Type ID), (Data Object Type, Data Object Type ID), (collection, Descriptor ID).
 - **Relation Description (1..N):** description of the relationship or role played by this Transfer Object Group Type with respect to the target object type or collection.

- **Relation type (1..1):** name for the relation (e.g., Data Entity Dictionary [DED], Syntax, Context, Provenance, Reference, Fixity Information) played by this object type or collection with respect to the target object type or collection.
- **Relation Textual Description (0..1):** text description of the relation.
- **Data Object Type (0..N):** attribute that supports a description of the Data Object Type. A Data Object Type describes one or more Data Objects (instances). There may be multiple Data Object Types within a Transfer Object Group Type.
 - **Data Object Type ID (1..1):** identifier of the Data Object Type. It shall be unique across the Producer-Archive Project. Uniqueness shall be checked by the Archive. This identifier may be used by a relationship attribute to refer to this Data Object Type.
 - **Data Object Type Description (0..1):** explanatory text describing the meaning of this Data Object Type.
 - **Data Object Type Occurrence (1..1):** number of instances (Data Objects) of this Data Object Type in each instance of its containing group. This may be expressed as one, or more, or as a range of values. This number may not be known at the time of descriptor creation. In the case of a unique value, the Minimal Occurrence and the Maximal Occurrence shall have the same value.
 - **Minimal Occurrence (1..1):** non-negative integer value (\leq Maximal Occurrence).

Exactly one of the two following attributes ‘Maximal Occurrence’ or ‘Maximal Unknown’ shall be used.

 - **Maximal Occurrence (1..1):** non-negative integer value (\geq Minimal Occurrence).
 - **Maximal Unknown (1..1):** empty attribute when used.
 - **Data Object Type File Occurrence (0..1):** number of file instances in each Data Object Type instance. This may be expressed as one, or more, or as a range of values. This number may not be known at the time of descriptor creation. In the case of a unique value, the Minimal Occurrence and the Maximal Occurrence shall have the same value. The absence of this attribute implies there shall be one file.
 - **Minimal Occurrence (1..1):** non-negative integer value (\leq Maximal Occurrence).

Exactly one of the two following attributes ‘Maximal Occurrence’ or ‘Maximal Unknown’ shall be used.

- **Maximal Occurrence (1..1):** non-negative integer value (\geq Minimal Occurrence).
- **Maximal Unknown (1..1):** empty attribute when used.
- **Data Object Type Format (0..1):** attribute used to identify the format of this Data Object Type in the Transfer Object Group Type using a MIME type or another registration mechanism.
 - **Mime Type (0..1):** attribute that identifies the MIME type for the Data Object Type (e.g., ‘application/octet-stream’ MIME Type as documented by the Internet Engineering Task Force [IETF]).
 - **Registration Information (0..1):** attribute that supports information used to identify the format of this Data Object Type as registered with a specified registration authority.
 - **Registration Authority (0..1):** identifier of the organization or system that has registered and holds the format description (e.g., IETF, CCSDS Control Authority[CA]).
 - **Registered ID (0..1):** identifier of the format description within the context of the registration authority (e.g., NSSD0145 under CCSDS CA).
- **Data Object Type Encoded (0..N):** characterization of how a Data Object has been encoded. May be used to describe possible nested encodings, for example a Data Object encoded in a zip file and then encoded in a binhex file.
 - **Encoding Name (1..1):** name for the type of encoding, such as zip, tar, or other name not included in this list.
 - **Encoding Description (1..1):** description or reference to the type of the encoding.

The encodings shall be described in the order of application.
- **Data Object Type Association (0..N):** attribute used to describe a relationship from this Data Object Type to the object type or collection having the identifier given by the Target ID attribute (targeted object types include Data Object Type, Transfer Object Type, Transfer Object Group Type). These are transversal links specifying the type of relation.
 - **Target ID (1..1):** identifier of target object type or collection of the Producer-Archive Project to which the relationship from these objects is established. Possible target object type and their corresponding identifiers include: (Transfer Object Type, Descriptor ID), (Transfer Object Group Type, Transfer Object Group Type ID), (Data Object Type, Data Object Type ID), (collection, Descriptor ID).

- **Relation Description (1..N):** description of the relationship or role played by this Data Object Type with respect to the target object type or collection.
 - **Relation Type (1..1):** name for the relation (e.g., Data Entity Dictionary [DED]), Syntax, Context, Provenance, Reference, Fixity Information) played by this object type or collection with respect to the target object type or collection.
 - **Relation Textual Description (0..1):** text description of the relation.
 - **Any (0..N):** mechanism that allows any new attribute(s) to be added as needed.
 - **Transfer Object Group Type (0..N):** attribute that may be used to describe each of the nested content groupings when the Transfer Object group consists of a hierarchical structure.
 - **Any (0..N):** mechanism that allows any new attribute(s) to be added as needed.
- **Implementation: XML Schema**

The Transfer Object Group Type attribute shall be implemented as the ‘transferObjectGroupType’ element as specified in the XML schema given in A2, and common types given in A1.

NOTES

- 1 Figure 3-6 is a schema diagram for the ‘transferObjectGroupType’ element.
- 2 As shown in figure 3-6, the ‘groupEncoded’ parent element is of type ‘encodingType’. This type is used as well in the ‘dataObjectTypeEncoded’ element. Also as shown in the figure, the ‘groupTypeAssociation’ element is made up of the ‘association’ element of type ‘associationType’. This type is used as well in the ‘dataObjectType’ element, and in the relation element of the ‘transferObjectTypeDescriptor’ and the ‘collectionDescriptor’.

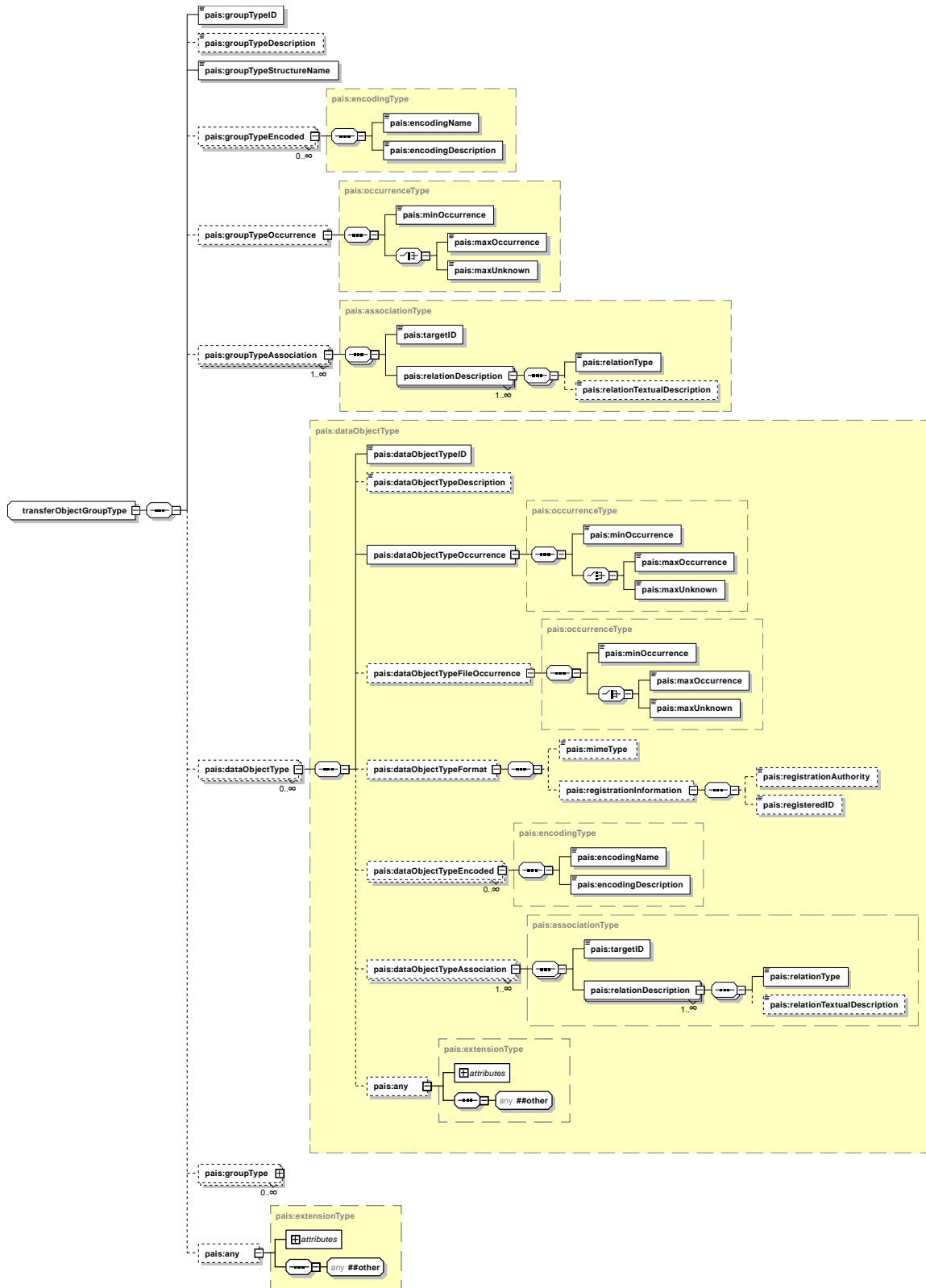


Figure 3-6: Transfer Object Type Descriptor 'groupType'

3.2.2.6 User Defined Attributes

– **Abstract View:**

Any (0..N): mechanism that allows a conforming XML document to have any additional attributes at this point in its structure.

– **Implementation: XML Schema**

The Any attribute shall be implemented as the ‘any’ element as specified in the XML schema given in A1.

NOTES

- 1 Figure 3-7 is a schema diagram for the ‘any’ element.
- 2 The ‘any’ element allows many XML attributes, and one single element. This element may be complex and thus may incorporate other additional elements.

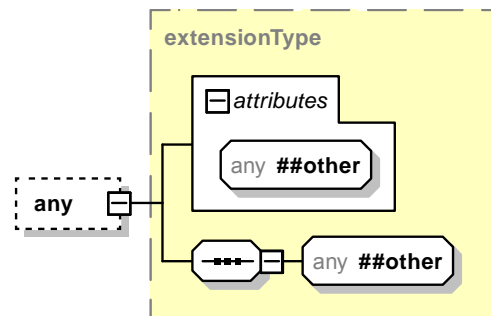


Figure 3-7: Transfer Object Type Descriptor ‘any’

3.3 COLLECTION DESCRIPTOR SPECIFICATION (CCSD0015)

3.3.1 TOP-LEVEL SPECIFICATION

3.3.1.1 Abstract View Top-Level Specification

3.3.1.1.1 The Collection Descriptor shall be composed of a set of attributes and values that are assigned to these attributes in order to characterize one or more collection views of the objects to be transferred.

NOTE – The Collection Descriptor does not describe the individual Transfer Object Types but rather gives a formal view of a collection of Transfer Objects (or collections of collections).

3.3.1.1.2 These attributes shall be grouped into the following top-level categories with the following definitions:

- **Descriptor Identification:** set of attributes supporting the unique identification of a collection within the Producer-Archive Project. Users may incorporate additional attributes as needed to further specialize the Descriptor Identification.
- **Collection Description:** set of attributes giving a high-level description of the collection, such as the title, the content, and the size. Users may incorporate additional attributes as needed to further specialize the Collection Description.
- **Collection Relationships:** set of attributes that identify directional relationships between this collection and other collections, Transfer Object Types, Transfer Object Group Types, or Data Object Types. These relationships are specified using identifiers as given within the Descriptors of the MOT. Users may incorporate additional attributes as needed to further specialize the Collection Relationships.
- **User Defined Attributes:** new attributes that users may incorporate as needed to further specialize the Collection Descriptor.

NOTES

- 1 In the following, each attribute has a mandatory or optional status, and has an occurrence. In the text, the occurrence is indicated between parentheses (min..max) after the name of the attribute; a zero value means that the attribute is optional, and mandatory otherwise.
- 2 The attributes themselves belong to groups that can be mandatory or optional. Inside an optional group, an attribute may be mandatory, meaning that the attribute must be present if there is an instance of that group. (On the contrary, inside a mandatory group, an attribute may be optional.)

3.3.1.2 XML Schema of Top-Level Specification

3.3.1.2.1 The Collection Descriptor shall conform to the complete XML schema given in A3, and common types given in A1.

3.3.1.2.2 The Collection Descriptor XML schema shall contain XML elements corresponding to the top-level categories as specified in the abstract view given above.

3.3.2 DISCUSSION

Figure 3-8 provides the schema diagram for this first-level decomposition of the Collection Descriptor. Figure 3-9 provides the schema diagram for the complete decomposition of the Collection Descriptor. Descriptor attributes are implemented as XML elements.

In the next section the semantics for each attribute in a category are given along with a view of the subset of the XML schema that applies.

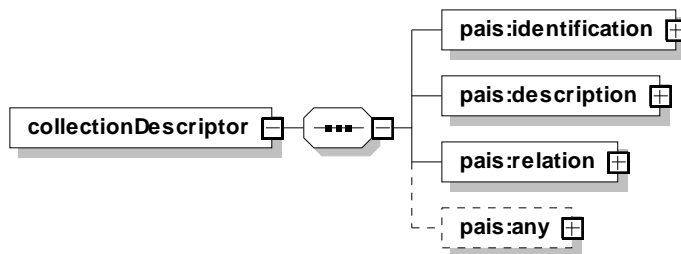


Figure 3-8: First Decomposition Level of ‘collectionDescriptor’

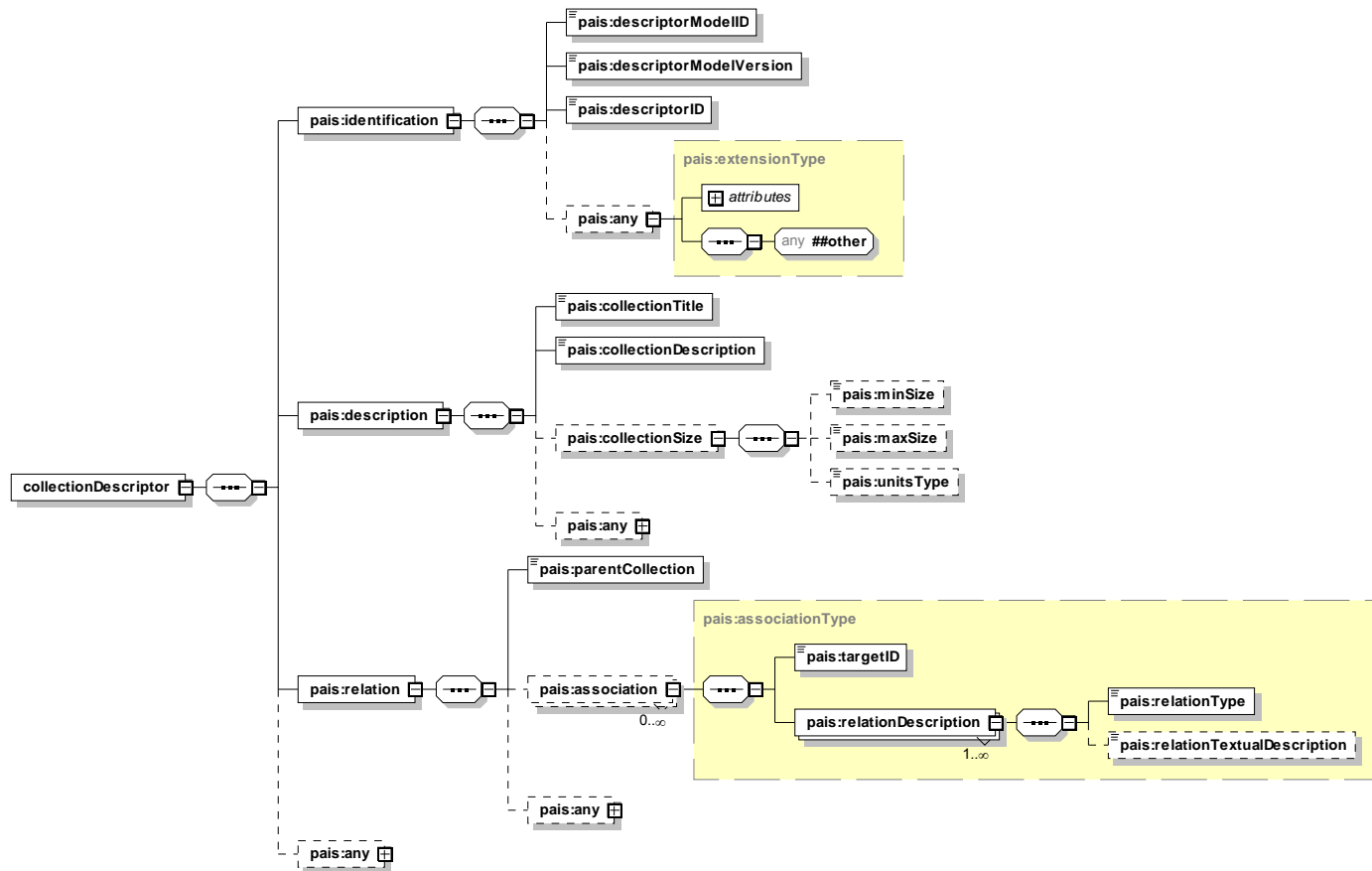


Figure 3-9: Complete Decomposition of a ‘collectionDescriptor’

3.3.3 DETAILED DESCRIPTION

3.3.3.1 General

3.3.3.1.1 Each attribute in the identified categories shall have the semantics as specified in the following subsections having the introductory phrase ‘Abstract view’.

3.3.3.1.2 Each attribute in the identified categories shall be implemented in accordance with the XML schemas as specified in the following subsections having the introductory phrase ‘Implementation: XML schema’.

3.3.3.2 Collection Identification

– **Abstract View:**

Identification (1..1)

- **Descriptor Model ID (1..1):** identifier of the Descriptor Model upon which this Collection Descriptor is based. It may be the Descriptor Model as given in the standard or it may be a specialized version as described in detail in 3.4. It shall be unique across the Producer-Archive Project and shall be provided by the Archive.

The standard value for this ID, when no specialization has been employed, is ‘CCSD0015’. When specialization has been employed, the value for the Descriptor Model ID should be agreed between the Producer and the Archive.

- **Descriptor Model Version (1..1):** identifier of the version of the Collection Descriptor Model. This allows tracking updates to the identified Descriptor Model.

The standard value for this ID, when the Descriptor Model ID is ‘CCSD0015’ and no revision to this standard model has occurred, is ‘V1.0’. When specialization has been employed, the value for the Descriptor Model Version shall be agreed between the Producer and the Archive.

- **Descriptor ID (1..1):** identifier of this Descriptor within the Producer-Archive Project. Uniqueness shall be checked by the Archive. This identifier may be used by a relationship attribute to refer to this collection. The highest-level Descriptor in the MOT shall be a Collection Descriptor. The value for its Descriptor ID shall be the Producer-Archive Project ID. The Producer-Archive Project ID shall be assigned by the Archive.
- **Any (0..N):** mechanism that allows any new attribute(s) to be added as needed.

– **Implementation: XML Schema**

The identification attributes shall be implemented as the ‘identification’ element as specified in the XML schema given in A3, and common types given in A1.

NOTE – Figure 3-10 is a schema diagram for the ‘identification’ element.

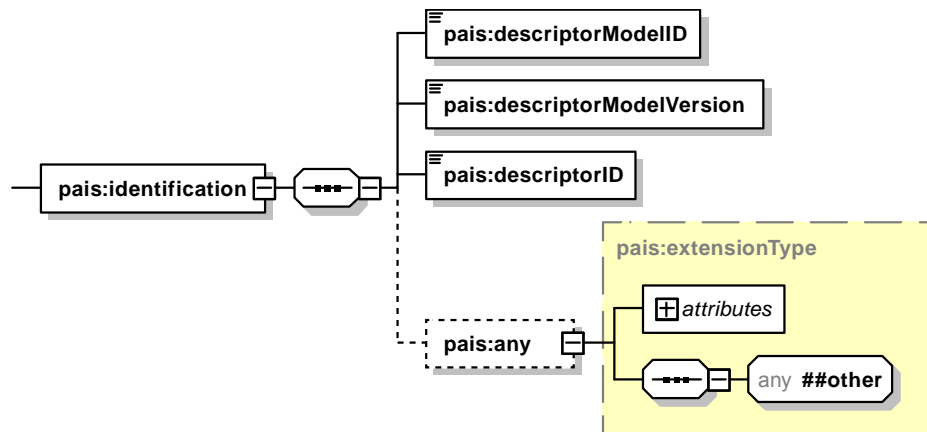


Figure 3-10: Collection Descriptor ‘identification’

3.3.3.3 Collection Description

– **Abstract View:**

Collection Description (1..1)

- **Collection Title (1..1):** extensive descriptive phrase used as the name of the collection.
- **Collection Description (1..1):** explanatory text describing the meaning of the collection.
- **Collection Size (0..1):** characterization of the size range of this collection. The size range may not be known at the time of descriptor creation and therefore the corresponding attribute may be omitted.
 - **Minimal Size (0..1):** float value (\leq maximal size when specified).
 - **Maximal Size (0..1):** float value (\geq minimal size when specified).
 - **Units Type (1..1):** list of units: KB, MB, GB, TB, PB.
- **Any (0..N):** mechanism that allows any new attribute(s) to be added as needed.

– **Implementation: XML Schema**

The identification attributes shall be implemented as the ‘description’ element as specified in the XML schema given in A3, and common types given in A1.

NOTE – Figure 3-11 is a schema diagram for the ‘description’ element.

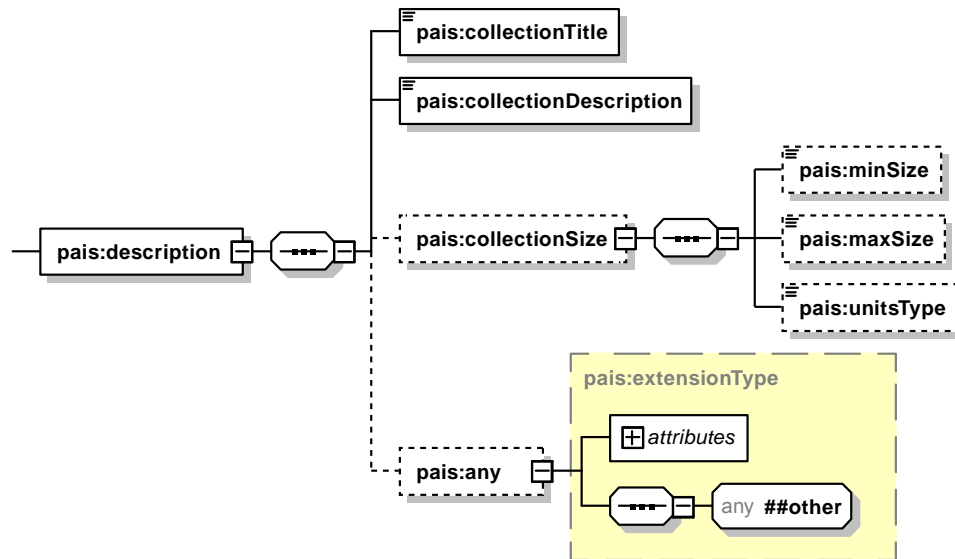


Figure 3-11: Collection Descriptor ‘description’

3.3.3.4 Collection Relationships

– Abstract view:

Relationships (1..1)

- **Parent Collection (1..1):** identifier of a Collection Descriptor that provides an aggregation view that includes this collection. The value for the parent collection of the highest-level Descriptor in the MOT shall be ‘none’.
- **Association (0..N):** attribute used to describe a relationship from this collection to the object type or collection having the identifier given by the Target ID attribute (targeted object type include Transfer Object Type, Transfer Object Group Type, and Data Object Type). These are transversal links specifying the type of relation.
 - **Target ID (1..1):** identifier of target object or collection of the Producer-Archive Project to which the relationship is established. Possible target object types and their corresponding identifiers include: (Transfer Object Type, Descriptor ID), (Transfer Object Group Type, Transfer Object Group Type ID), (Data Object Type, Data Object Type ID), (collection, Descriptor ID).
 - **Relation Description (1..N):** description of the relationship or role played by this collection with respect to the target object type or collection.
 - **Relation Type (1..1):** name for the relation (e.g., Data Entity Dictionary [DED], Syntax, Context, Provenance, Reference, Fixity Information) played by this collection with respect to the target object type or collection.
 - **Relation Textual Description (0..1):** text description of the relation.

- **Any (0..N):** mechanism that allows any new attribute(s) to be added as needed.

– **Implementation: XML Schema**

The Relation attributes shall be implemented as the ‘relation’ element as specified in the XML schema given in A3, and common types given in A1.

NOTE – Figure 3-12 is a schema diagram for the ‘relation’ element.

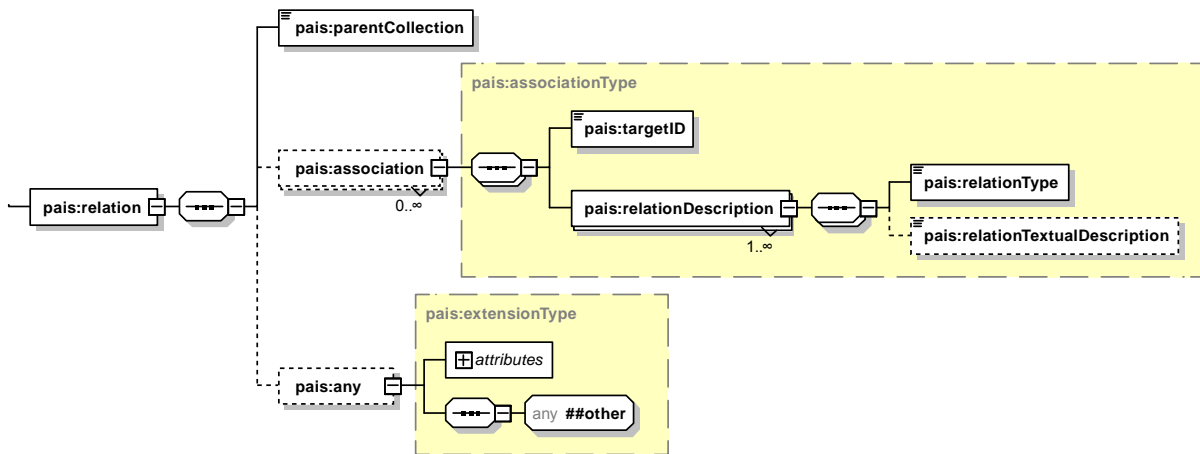


Figure 3-12: Collection Descriptor ‘relation’

3.3.3.5 User Defined Attributes

– **Abstract view:**

Any (0..N): mechanism that allows a conforming XML document to have any additional attributes at this point in its structure.

– **Implementation: XML schema**

The Any attribute shall be implemented as the ‘any’ element as specified in the XML schema given in A1.

NOTE – The ‘any’ element allows many XML attributes, and one single element. This element may be complex and thus may incorporate other additional elements.

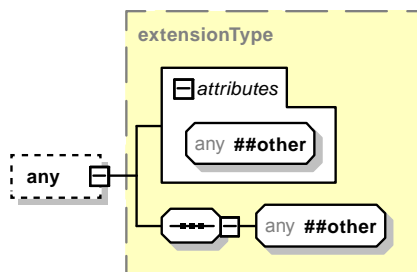


Figure 3-13: Collection Descriptor ‘any’

3.4 SPECIALIZATION OF THE DESCRIPTOR MODELS

3.4.1 OVERVIEW

The previous Models (Transfer Object Type Descriptor Model and Collection Descriptor Model) are intended to cover the different possible situations and contain a small set of mandatory attributes, some additional optional attributes, and the ability to add User Defined attributes in each section.

These Descriptor Models may be specialized to achieve the applicable level of description desired by the Producer-Archive Project (see figure 3-14):

- Specialization of a domain Model from the standard Model.
- Domain specialization for creating a Producer-Archive Project Model (other intermediary levels may be considered if necessary). Producer-Archive Project Descriptor Models may be created directly from the standard Descriptor Models.

The Producer-Archive Project Descriptor Models are discussed by the Archive and the Producer during the Formal Definition Phase.

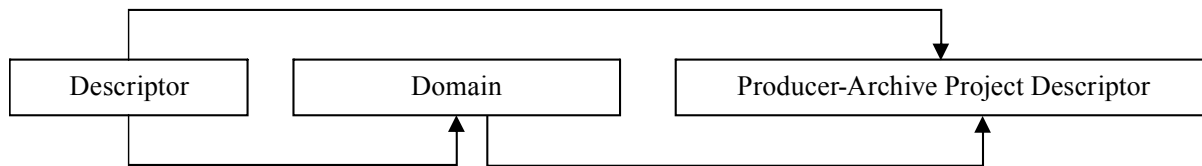


Figure 3-14: Specialization of the Descriptor Models

3.4.2 ABSTRACT SPECIALIZATION OF DESCRIPTOR MODELS

Abstract specialization of the standard models may only be accomplished through the following modifications:

- addition of ‘User Defined’ attributes;
- elimination of optional attributes;
- modification of existing attributes: by occurrence restriction, by modification of status attribute (e.g., optional changed to mandatory), by text restriction (use of list). The name of an attribute should not be modified (to adapt to the Producer-Archive Project terminology for example), in order that the tools manipulating Descriptors remain usable.

3.4.3 CONCRETE SPECIALIZATION OF THE DESCRIPTOR MODELS

3.4.3.1 A specialized Descriptor Model shall be an XML document that conforms to an XML schema derived from the XML schema of a previously defined Descriptor Model.

3.4.3.2 If specialization of a Descriptor Model is required, its XML schema may only be specialized through the following modifications:

- addition of new elements (or sequence of elements) by the use of the element ‘any’;
- suppression of optional elements;
- optional elements turned to mandatory;
- modification of occurrence numbers (for example 10..15 instead of 1..n);
- definition of a list (enumeration) for strings (for example ‘groupStructureName’ =limited list of ‘set’, ‘sequence’, ‘directory’, ‘undescribed’);
- definition of patterns for identifiers (imposed nomenclature) or restriction of string length;
- restriction of type ‘nonNegativeInteger’ for number of occurrences to a sub type that is acceptable to both the Producer and the Archive (for example ‘int, short’).

4 DEFINITION OF CONSTRAINTS ON TYPES OF SIPS (CCSD0016)

4.1 OVERVIEW

Two types of constraints apply to SIPS: the first one defines the authorized content of each SIP; the second one defines the order in which SIPS have to be delivered by the Producer to the Archive. The constraints are associated with a specific Producer-Archive Project.

4.2 ABSTRACT SIP CONSTRAINTS

4.2.1 GENERAL

The SIP constraints shall be defined by the following set of attributes and values that are assigned to these attributes in order to characterize the constraints.

- **Producer-Archive Project Identifier (1..1)**: a unique identifier of the Producer-Archive Project among all Producer-Archive Projects undertaken by this Archive. This ID shall be provided by the Archive for use in the SIPS (see also first paragraph section 3).

NOTE – An overall table of identifiers and their management can be found in annex D.

- **SIP Content Type (1..N)**: a set of attributes describing the content of each type of SIP.
- **SIP Sequencing Constraint Group (0..N)**: a set of attributes describing the delivery order applying to the SIP Content Types within the Producer-Archive Project. There are no constraints between the SIPS belonging to different SIP Sequencing Constraint Groups.

4.2.2 ABSTRACT SIP CONTENT TYPES SPECIFICATION

4.2.2.1 Each Producer-Archive Project shall define one or more SIP Content Types.

NOTE – A SIP Content Type identifies the Transfer Object Types that may be found within that type of SIP. That type of SIP is itself identified with a SIP Content Type Identifier, which is unique within the Producer-Archive Project.

4.2.2.2 The types of SIPS shall be defined by the following attributes:

- **SIP Content Type (1..N)**
 - **SIP Content Type ID (1..1)**: identifier of the SIP Content Type. It shall be unique across the Producer-Archive Project. Uniqueness shall be checked by the Archive. The SIP Content Type ID shall be referenced in each created SIP.

- **Authorized Descriptor (1..N):** a list of the Transfer Object Type Descriptor Identifiers that a transferred SIP of the type ‘SIP Content Type ID’, may or may not contain. A transferred SIP shall not contain a Transfer Object associated with a Descriptor ID that is not in this list.
 - **Transfer Object Type Descriptor ID (1..1):** identifier for this Transfer Object Type which is allowed in this type of SIP.
 - **Occurrence (1..1):** number of instances (Transfer Objects) of this Transfer Object Type that will be present in a SIP of this type. This may be expressed as one, or more, or as a range of values. If the number of Transfer Objects is not known at the time of SIP Content Type specification creation, then zero shall be specified as the Minimal Occurrence and the Maximal Unknown attribute shall be used. In the case of a unique value, the Minimal Occurrence and the Maximal Occurrence shall have the same value.
 - **Minimal Occurrence (1..1):** non-negative integer value (\leq maximal occurrence).

Exactly one of the two following attributes ‘Maximal Occurrence’ or ‘Maximal Unknown’ shall be used.

 - **Maximal Occurrence (1..1):** non-negative integer value (\geq minimal occurrence).
 - **Maximal Unknown (1..1):** empty attribute if chosen.

4.2.3 ABSTRACT SIP SEQUENCING CONSTRAINTS SPECIFICATION

4.2.3.1 Each Producer-Archive Project may define the sequencing constraints that must apply to the transferred SIPs.

NOTE – Sequencing constraints are specified using SIP Content Types and therefore a given sequencing constraint applies to all SIPs of the referenced SIP Content Types.

4.2.3.2 If there are no sequencing constraints, the SIPs may be transferred independently of each other in any order.

4.2.3.3 SIP Sequencing constraints shall be defined by the following attributes:

- **SIP Sequencing Constraint Group (0..N):** a set of attributes describing the delivery order applying to the SIP Content Types within the Producer-Archive Project. There are no constraints between the SIPs belonging to different SIP Sequencing Constraint Groups.
 - **Group Name (0..1):** name for this group of constraint items.
 - **Constraint Item (2..N):**

- **SIP Content Type ID (1..1):** identifier of a SIP Content Type. It identifies each of the types of SIPs that are under constraints in the corresponding group.
- **Constraint Serial Number (1..1):** a number that specifies the order of delivery of SIPs with the associated SIP Content Type identifier. All SIPs of a SIP Content Type associated with a Constraint Serial Number with a value of ‘n’ must be delivered before all SIPs of a SIP Content Type associated with a Constraint Serial Number with a value greater than ‘n’. There are no constraints between two or more SIP Content Types with the same Constraint Serial Number value inside the same group.

NOTE – Sequencing constraints are specified only at the SIP Content Type level and cannot be specified for Transfer Object Types or other components of SIP Content Types.

4.3 SIP SEQUENCING CONSTRAINTS IMPLEMENTATION

The SIP Sequencing Constraints implementation shall conform to the XML schema given in A4, and common types given in A1.

NOTE – This schema is shown schematically in figure 4-1. It defines elements describing the abstract SIP Constraints defined in 4.2 including the SIP Content Types defined in 4.2.2 and the SIP sequencing constraints defined in 4.2.3.

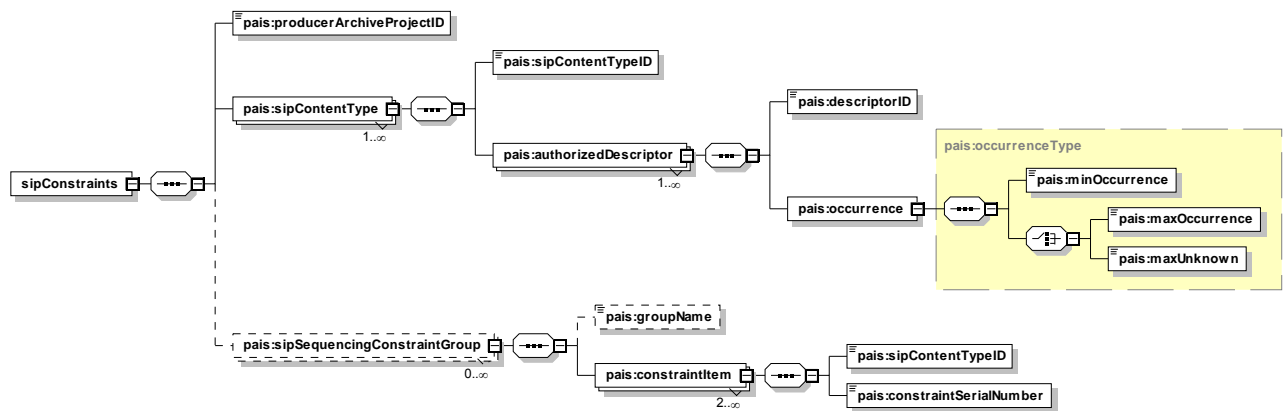


Figure 4-1: Decomposition of ‘sipConstraints’

5 ABSTRACT SIP SPECIFICATION (CCSD0017)

5.1 OVERVIEW

SIPs carry the data, or pointers to the data, being transferred to an Archive. The abstract SIP, or SIP Model, is an abstraction that puts constraints on all possible SIPs. It conceptually conveys one or more complete Transfer Objects. It also conceptually conveys a number of attributes about the SIP.

The framework for this SIP model is based on the concept of containers. The SIP Model is a container that holds any number of internal containers which themselves may have containers, and so on, thus supporting multiple hierarchies of containers. A container may also hold attributes about itself.

5.2 SIP MODEL SPECIFICATION

5.2.1 The following SIP Model shall be used to put constraints on the implementation of SIPs for the transfer of data.

5.2.2 At the highest level, the SIP shall consist of three types of containers as shown schematically in figure 5-1. There is

- one SIP Global Information container;
- any number of Transfer Object containers; and
- any number of Transfer Object To Delete containers.

5.2.3 Each container and each attribute in the identified containers below shall have the semantics as specified.

NOTES

- 1 In the following, each container or attribute has a mandatory or optional status, and has an occurrence. In the text, the occurrence is indicated between parentheses (min..max) after the name of the container or attribute; a zero value means that it is optional, and mandatory otherwise.
- 2 The containers or attributes themselves belong to containers that can be mandatory or optional. Inside an optional container, an attribute may be mandatory meaning that the attribute must be present if there is an instance of that container. (On the contrary, inside a mandatory container, an attribute may be optional.)

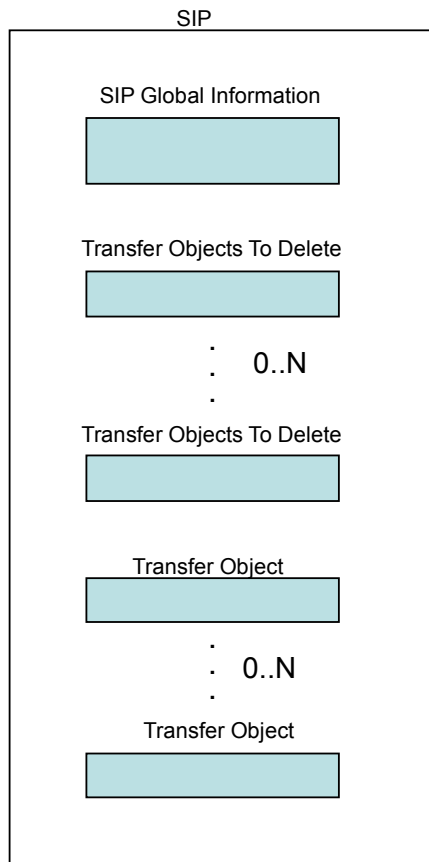


Figure 5-1: Abstract View of the SIP Container

5.2.4 The content of these containers shall be as follows:

- **SIP Global Information container (1..1):** SIP structure that holds a set of attributes supporting the unique identification of each SIP within the Producer-Archive Project and the ability to optionally track the sequencing of SIPs. Users may incorporate additional attributes as needed to further specialize the SIP Global Information.
 - **SIP ID (1..1):** identifier of the delivered SIP within the context of the given Producer-Archive Project. If there are multiple Producer Sources submitting SIPs within a single Producer-Archive Project, this SIP ID must be unique across all such Producer Sources. It is inserted during SIP construction. The form shall be agreed between Producer and Archive, but the identifier shall be generated by the Producer. The Archive shall check uniqueness.
 - **Producer-Archive Project ID (1..1):** identifier of the Producer-Archive Project that distinguishes the project from all other Producer-Archive Projects undertaken by this Archive. This ID shall be provided by the Archive for use in the SIPs (see also first paragraph section 3).
 - **Producer Source ID (1..1):** identifier of the Producer Source sufficient to enable the Archive to identify the origin of the SIP and to send an acknowledgement.

There may be multiple Producer sources sending SIPs to the Archive within a given Producer-Archive Project. The form of the Producer Source ID shall be agreed between Producer and Archive.

- **SIP Content Type ID (1..1)**: identifier of the specification that defines which Transfer Object Types (i.e., Descriptor IDs) are allowed within this SIP, as well as their occurrence within the SIP. It has been defined previously in this document (see section 4).
 - **SIP Sequence Number (0..1)**: number indicating the order in which the SIP has been sent. This number is unique within the combined context of the Producer-Archive Project and Producer Source ID. This becomes mandatory for all SIPs sent by a Producer Source if any of the Transfer Objects to be provided by the Producer Source have a Descriptor that does not specify a unique value for the number of Transfer Objects to be delivered. It shall be generated by the Producer.
 - **Any (0..N)**: mechanism that allows a SIP to have any additional attributes within the structure of the SIP corresponding to this container.
- **Transfer Objects To Delete container (0..N)**: SIP structure composed of one or more attributes giving the identification of the Transfer Objects previously sent to the Archive that must be deleted by the Archive. Users may incorporate additional attributes as needed to further specialize the Transfer Object To Delete container.
 - **Transfer Object To Delete ID (1..N)**: identifier of the Transfer Object ID of a previously sent Transfer Object that is to be deleted by the Archive.
 - **Any (0..1)**: mechanism that allows this SIP structure to have any additional attributes that may be needed.
 - **Transfer Object container (0..N)**: SIP structure that conceptually holds two types of containers as shown schematically in figure 5-2. The Transfer Object container consists of:
 - one Transfer Object Identification and Status container; and
 - one or more Transfer Object Group containers.

NOTE – Users may incorporate additional attributes as needed to further specialize the Transfer Object container.

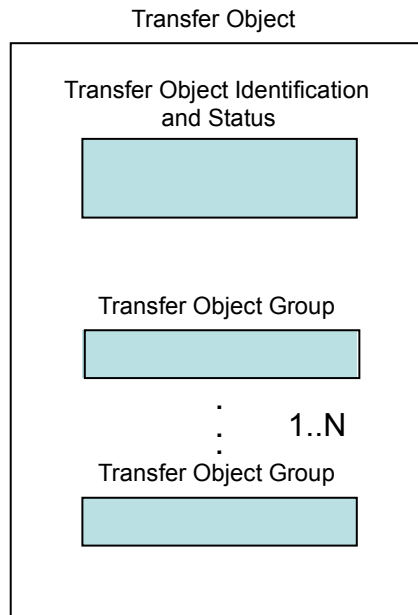


Figure 5-2: Abstract View of the Transfer Object Container

The content of these two types of containers shall be as follows:

- **Transfer Object Identification and Status container (1..1):** SIP structure that holds a set of attributes supporting unique identification and replacement status information about this Transfer Object in the SIP. Users may incorporate additional attributes as needed to further specialize the Transfer Object.
 - **Descriptor ID (1..1):** identifier of the Transfer Object Type Descriptor that describes this type of Transfer Object. This is obtained from the MOT.
 - **Transfer Object ID (1..1):** identifier for each delivered Transfer Object within the Producer-Archive Project. It is inserted during SIP building. The form shall be agreed between Producer and Archive. For example, it could be constructed by concatenating the SIP ID (or Descriptor ID) and some sequence number for each Transfer Object. It shall be generated by the Producer. Uniqueness shall be checked by the Archive.
 - **Last Transfer Object Flag (0..1):** indicator specifying that this Transfer Object is the last Transfer Object of this type (i.e., within the scope of this Descriptor) being delivered by this Producer Source. This attribute is particularly useful when the number of Transfer Objects to be delivered is not known in advance. If used with a single Producer Source for Transfer Objects of this type, this flag eliminates the need for an additional contact between the Archives and the Producer Source to verify that all such Transfer Objects have been received. If there are multiple Producer Sources that may be delivering Transfer Objects of this type, the Archive may or may not need to

contact these Producer Sources to determine when all such Transfer Objects have been sent and received.

- **Replacement Transfer Object ID (0..1)**: identifier of the Transfer Object ID of a previously sent Transfer Object that is to be replaced by this Transfer Object.
- **Any (0..N)**: mechanism that allows this SIP structure to have any additional attributes that may be needed.
- **Transfer Object Group container (1..N)**: SIP structure that conceptually holds three types of containers as shown schematically in figure 5-3. These are the
 - Transfer Object Group Identification container;
 - any number of other Transfer Object Group containers; and
 - any number of Data Object containers.

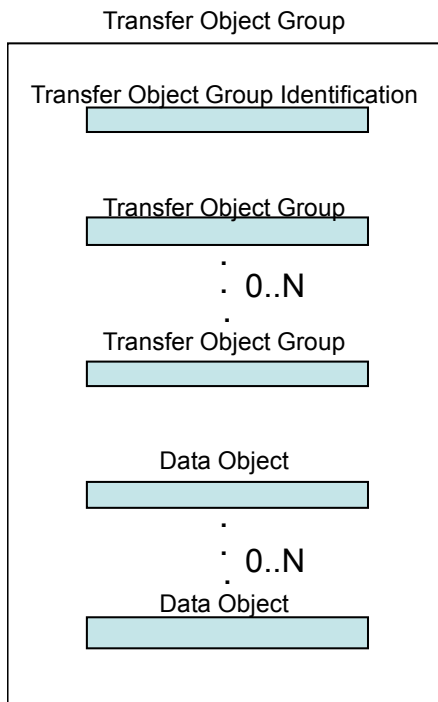


Figure 5-3: Abstract View of the Transfer Object Group Container

One or more instances of a Transfer Object Group container within a SIP shall result from the presence of each Transfer Object Group Type described within the associated Transfer Object Type Descriptor, except for two special cases.

- Case 1

The Transfer Object Group Type has a specified encoding that converts the group into a single file. Regardless of the values for other attributes in the Transfer Object Group Type in the Descriptor, the result shall be a single Data Object container within a SIP for each instance of such a Transfer Object Group Type in the Descriptor.

- Case 2

The Transfer Object Group Type has a Transfer Object Group Type Structure Name whose value is 'undescribed'.

- If this type also has specified an encoding, again there shall be a single Data Object container within a SIP for each instance of such a Transfer Object Group Type in the Descriptor.
- If this type has not specified an encoding, the 'undescribed' Transfer Object Group Type in the Descriptor shall result, in the SIP, in some combination of Transfer Object Group containers and Data Object containers for each instance of such a Transfer Object Group Type in the Descriptor. The Data Object containers shall be used for the files and the Transfer Object Group containers shall be used for directories or for conceptually holding multiple Data Object containers.

The content of these three types of containers shall be as follows:

- **Transfer Object Group Identification container (1..1):** SIP structure that holds a set of attributes identifying the type of group and optionally naming the group instance. Users may incorporate additional attributes as needed to further specialize the Transfer Object Group Identification information.
 - **Associated Descriptor Group Type ID (1..1):** identifier of the associated group description within the associated Descriptor.
 - If this group is an instance of a Transfer Object Group Type as specified in the Descriptor, then the value of this attribute shall be the Transfer Object Group Type ID of that Transfer Object Group Type.
 - If this group is an instance that is part of a data structure transferred under a Transfer Object Group Type whose Transfer Object Group Structure Name has the value 'undescribed', then the value of this Associated Descriptor ID shall be the Transfer Object Group Type ID of that Transfer Object Group Type.
 - Choice of one of the following two attributes (0..1):
 - **Transfer Object Group Instance Name (1..1):** name given to the group, such as a directory name, that is associated with the Transfer Object Group instance. It shall be provided by the Producer.

If the group has been modeled as a directory (i.e., Structure Name = ‘directory’), it shall be the name of the directory excluding any path information.

- **Transfer Object Group Preservation Name (1..1):** name given to the group, such as a directory name, that shall be preserved by the Archive in association with the Transfer Object Group instance. It shall be provided by the Producer.

If the group has been modeled as a directory (i.e., Structure Name = ‘directory’), it shall be the name of the directory excluding any path information.

- **Any (0..N):** mechanism that allows this SIP structure to have any additional attributes that may be needed.
- **Transfer Object Group container (0..N):** SIP structure that conceptually holds any number of additional Transfer Object Group containers.
- **Data Object Container (0..N):** SIP structure that conceptually holds two or more containers as shown schematically in figure 5-4. These are the:
 - Data Object Identification container; and
 - One or more Byte Stream containers.

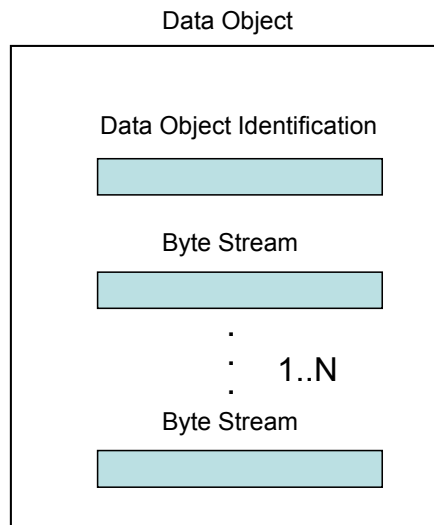


Figure 5-4: Abstract View of the Data Object Container

The content of these two types of containers shall be as follows:

- **Data Object Identification container (1..1):** SIP structure that holds a set of attributes identifying the type of Data Object and optionally supplying a name that is to be preserved along with the associated byte stream or streams. Users

may incorporate additional attributes as needed to further specialize the Data Object Identification information.

- **Associated Descriptor Data ID (1..1):** identifier of the associated data description within the associated Descriptor.
 - If this is an instance of a Data Object Type defined in the Descriptor then this shall be the Data Object Type ID of that Data Object Type.
 - If this is an instance of a Transfer Object Group Type defined in the Descriptor to be encoded and thus it results in a single Data Object, then this shall be the Transfer Object Group Type ID of that Transfer Object Group Type.
 - If this is an instance of a Data Object that is transferred within the context of a Descriptor defined Transfer Object Group Type whose Transfer Object Group Type Structure Name has the value ‘undescribed’, then this shall be the Transfer Object Group Type ID of that Transfer Object Group Type.
- **Data Object Preservation Name (0..1):** name to be preserved in association with the Data Object instance. When the Data Object is composed of a single byte stream, it tells the Archive exactly what name is to be preserved in association with that byte stream. When the Data Object is composed of multiple byte streams, it tells the Archive exactly what name is to be preserved in association with the set of byte streams. It shall be provided by the Producer.
- **Any (0..N):** mechanism that allows this SIP structure to have any additional attributes that may be needed.
- **Byte Stream container (1..N):** SIP structure that holds a set of attributes that provide a byte stream and/or a pointer to a byte stream outside the SIP.
 - **Byte Stream (0..1):** stream of bytes.
 - **Byte Stream Checksum (0..1):** checksum covering the stream of bytes.
 - **Pointer to Byte Stream (0..1):** a pointer to a byte stream outside the SIP.
 - **Any (0..N):** mechanism that allows this SIP structure to have any additional attributes that may be needed.

5.3 Discussion

A summary of the SIP container relationships described above is shown schematically in figure 5-5.

As noted above, an actual SIP may not include all of the possible container types and attributes provided by the SIP Model. For example, each Transfer Object will reflect the modeled view as given by the corresponding Descriptor. Therefore it may or may not have Transfer Object Group containers or Data Object containers under an ‘Undescribed’ Transfer Object Group container.

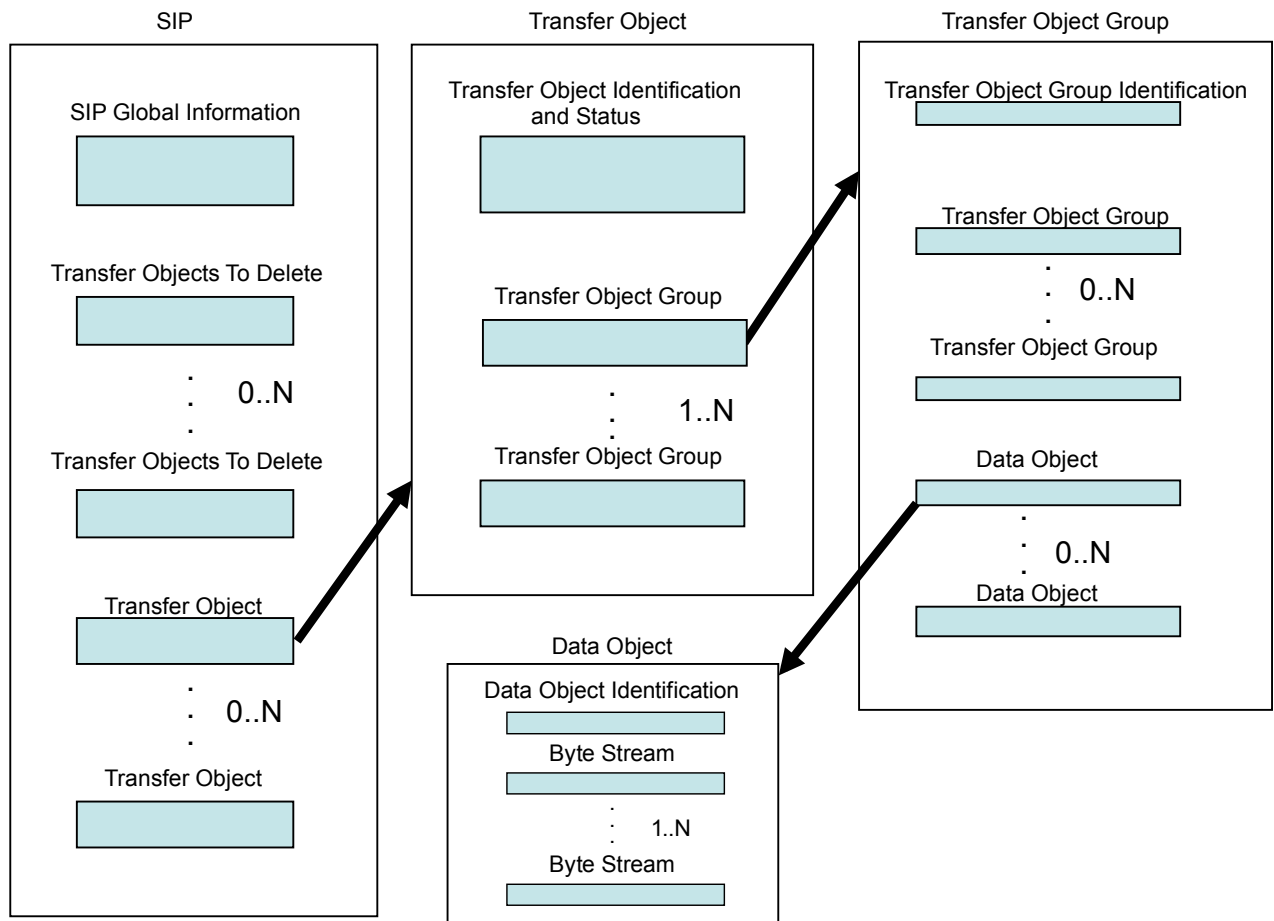


Figure 5-5: Abstract View of SIP, Transfer Object, Transfer Object Group, and Data Object

Additional constraints on actual SIPs will depend on the application of the SIP Constraints of section 4 which may limit the number and types of Transfer Objects present in a given SIP.

Validation of a Transfer Object in a SIP is facilitated by attributes in the Transfer Object containers that identify the relevant parts of the corresponding Descriptor that model the data. Thus:

- Each Transfer Object carries the ID of the corresponding Descriptor (Descriptor ID).
- Each Transfer Object Group carries the ID of the corresponding Descriptor section (Transfer Object Group Type ID).

- Each Data Object also carries the ID of the corresponding Descriptor section (Data Object Type ID or a Transfer Object Group Type ID of an ‘undescribed’ group which contains this Data Object or a Transfer Object Group Type ID of an ‘encoded’ group where the encoding results in a single file holding the entire group structure).
- Validation can also be facilitated by ensuring integrity of the byte streams through use of the optional checksum attribute.

5.4 SIP MODEL SPECIALIZATION

5.4.1 OVERVIEW

The SIP Model in this document is intended to support the requirements for the implementation of most SIPs. It defines a small set of mandatory attributes, some additional optional attributes, and the ability to add User Defined attributes in each section.

The SIP Model may be specialized to achieve the applicable level of SIP functionality and constraints desired by the Producer-Archive Project (see figure 5-6) through the following modifications:

- A Domain SIP Model may be created from the standard SIP Model in this document.
- A Producer-Archive Project SIP Model may be created directly from the standard SIP Model in this document or it may be created from a Domain SIP Model (or another intermediary level if necessary).

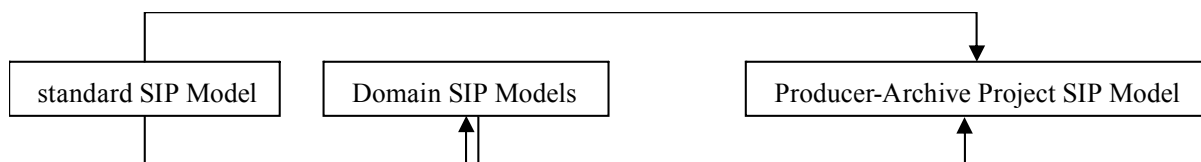


Figure 5-6: Specialization of the SIP Model

5.4.2 SPECIALIZATION

5.4.2.1 Any specialization of the SIP Model for a given Producer-Archive Project shall be discussed by the Archive and the Producer.

5.4.2.2 The specialization of the SIP Model for a given Producer-Archive Project shall be documented in any manner agreed by the Producer and the Archive.

5.4.2.3 To create a specialization, the base SIP Model shall only undergo the following modifications:

- addition of ‘User Defined’ attributes, if ‘User Defined’ attributes were allowed in the base SIP Model;

- elimination of optional attributes;
- modification of existing attributes: by occurrence restriction, by modification of status attribute (e.g., optional changed to mandatory), or by text restriction (use of list). The name of an attribute should not be modified (to adapt to the Producer-Archive Project terminology for example), in order that the tools manipulating SIPs remain usable.

6 SIP IMPLEMENTATION

6.1 OVERVIEW

A variety of SIP implementations are possible. In this document, a particular XFDU implementation is defined. SIP implementations in formats other than XFDU are allowed. They will not be considered 'XFDU PAIS SIP Conformant', but they could be 'Abstract SIP Conformant'. It is possible that additional PAIS SIP implementations may be standardized in the future.

6.2 SIP IMPLEMENTATION USING XFDU

6.2.1 OVERVIEW

This mapping of an abstract SIP to a SIP implementation is based on the XFDU Recommended Standard (see reference [1]).

The XFDU is a packaging standard that usually (but not necessarily) makes use of a zip file holding an XFDU-defined, xml-based, manifest file that conforms to the XFDU schema. For the purpose of conveying a SIP, the XFDU manifest has two key features of interest. These are the Package Header's environmentInfo element and the Content Unit element. Each of these incorporates the ability for users to add elements through the XFDU's extension element. The Content Unit element also allows nesting and thus supports annotations of hierarchical views. It also supports a pointer that resolves to a location of a file either within the internal content or externally.

6.2.2 GENERAL

The mapping of the abstract SIP to an XFDU PAIS SIP shall be as follows:

- a) A SIP shall be an XFDU instance.
- b) A SIP Global Information container shall be conveyed by the XFDU environmentInfo extension element.
- c) A SIP Transfer Object To Delete container shall be conveyed by a top-level XFDU Content Unit element that uses the associated XFDU extension element to convey the Transfer Object To Delete container attributes. If there are multiple Transfer Object To Delete containers, then there shall be a corresponding number of top-level XFDU Content Units.
- d) A SIP Transfer Object container shall be conveyed by a top-level XFDU Content Unit element that uses the associated XFDU extension element to convey the Transfer Object Identification and Status container attributes.
- e) A SIP Transfer Object Group container shall be conveyed by an XFDU Content Unit nested within the top-level Content Unit for the Transfer Object, or nested within the

Content Unit for another Transfer Object Group. The XFDU Content Unit for a given Transfer Object Group shall use the associated XFDU extension element to convey the Transfer Object Group Identification attributes. If the Descriptor's Transfer Object Group includes the encoding attribute, then there shall be no corresponding SIP Transfer Object Group container. Instead the SIP shall implement this as a SIP Data Object container.

- f) A SIP Data Object container shall be conveyed by an XFDU Content Unit nested within the XFDU Content Unit associated with the Transfer Object Group of which it is a part. The XFDU Content Unit of a given Data Object shall use the associated XFDU extension element to convey the Data Object Identification attributes.
- g) A SIP Byte Stream container shall be conveyed by an XFDU dataObjectPointer element nested within the XFDU Content Unit associated with the Data Object of which it is a part.

NOTE – Using an XFDU's dataObject element, the XFDU dataObjectPointer resolves to an XFDU byteStream element that conveys the SIP Byte Stream container's information. A Data Object with multiple byte streams will appear as a single XFDU Content Unit with multiple dataObjectPointers.

6.2.3 SIP XML SCHEMA FOR XFDU

6.2.3.1 The XFDU SIP implementation shall conform to the XML schema given in A5, and common types given in A1, and to the XFDU schema as given in reference [1].

NOTE – This schema defines elements describing the abstract SIP defined in section 5. This schema defines elements for use in the XFDU manifest and thus enables the XFDU to serve as a SIP. This high-level relationship is shown in figure 6-1.

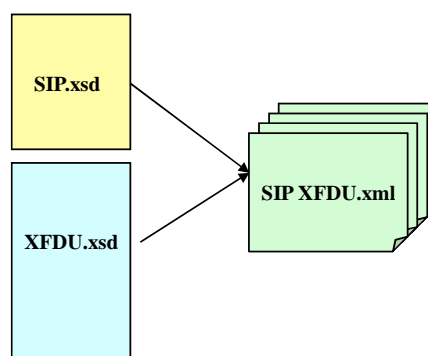


Figure 6-1: SIP and XFDU Schemas Constrain the XFDU XML Manifest to Implement the SIP as an XFDU

6.2.3.2 The following list of elements implemented in the SIP schema shall map to the corresponding attribute from section 5 as shown by the attribute in parenthesis:

- information associated with the SIP Global Information container:

- sipID (SIP ID),
- producerSourceID (Producer Source ID),
- producerArchiveProjectID (Producer-Archive Project ID),
- sipContentTypeID (SIP Content Type ID),
- sipSequenceNumber (SIP Sequence Number),
- any (Any, User defined attributes);

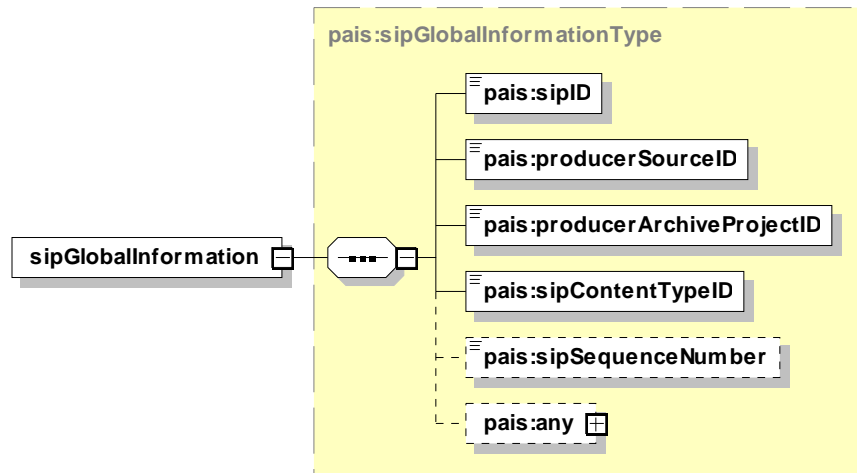


Figure 6-2: SIP Global Information

- information associated with each Transfer Object Identification and Status container:
 - descriptorID (Descriptor ID),
 - transferObjectID (Transfer Object ID),
 - lastTransferObjectFlag (Last Transfer Object Flag),
 - replacementTransferObjectID (Replacement Transfer Object ID),
 - any (Any, User defined attributes);

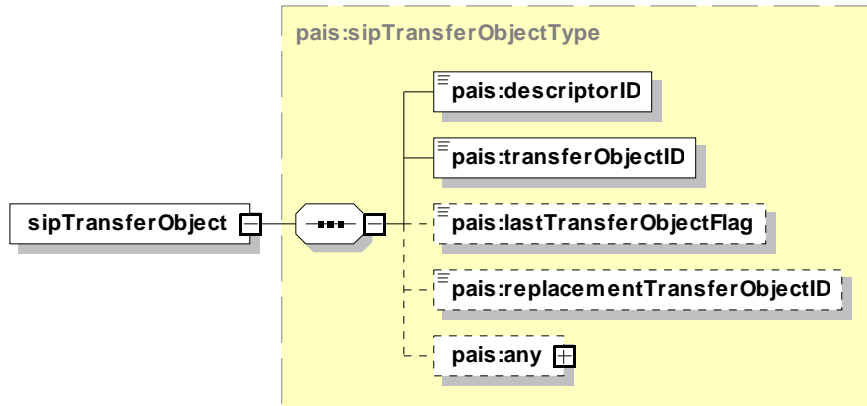


Figure 6-3: SIP Transfer Object Identification and Status

- information associated with each Transfer Object Group Identification container:
 - associatedDescriptorGroupTypeID (Associated Descriptor Group Type ID),
 - transferObjectGroupInstanceName (Transfer Object Group Instance Name),
 - transferObjectGroupPreservationName (Transfer Object Group Preservation Name),
 - any (Any, User defined attributes);

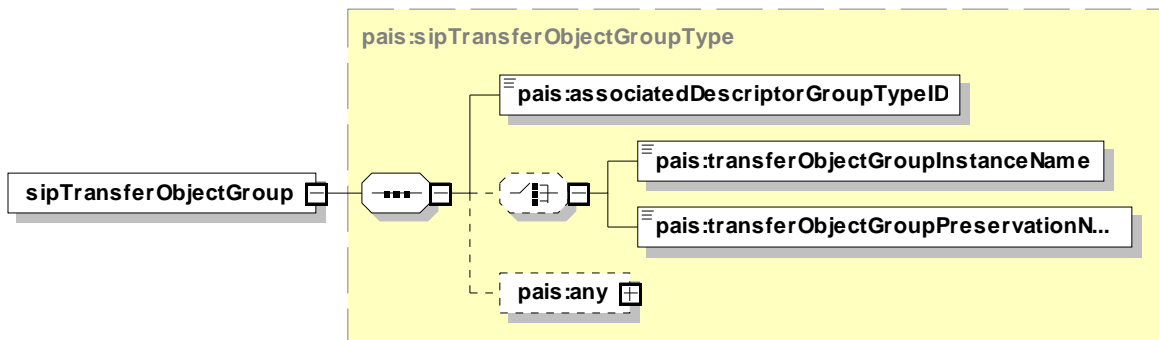


Figure 6-4: SIP Transfer Object Group Identification

- information associated with each embedded Data Object Identification container:
 - associatedDescriptorDataID (Associated Descriptor Data ID),
 - dataObjectPreservationName (Data Object Preservation Name),
 - any (Any, User defined attributes);

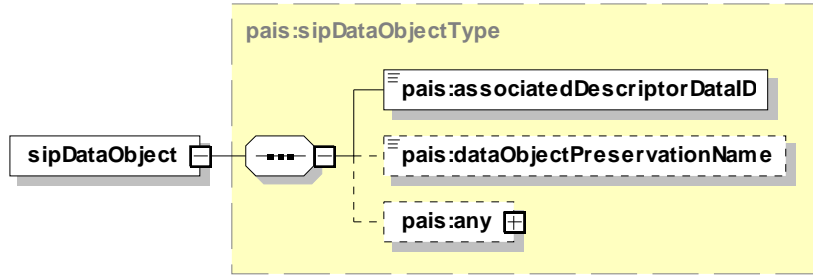


Figure 6-5: SIP Data Object Identification

- information associated with a Byte Stream container;

NOTE - A SIP schema supporting the Byte Stream container is not needed because the XFDU schema already contains the needed elements as discussed in 6.2.4.

- information associated with a Transfer Object to Delete container:
 - transferObjectToDeleteID (Transfer Object To Delete ID),
 - any (Any, User defined attributes);

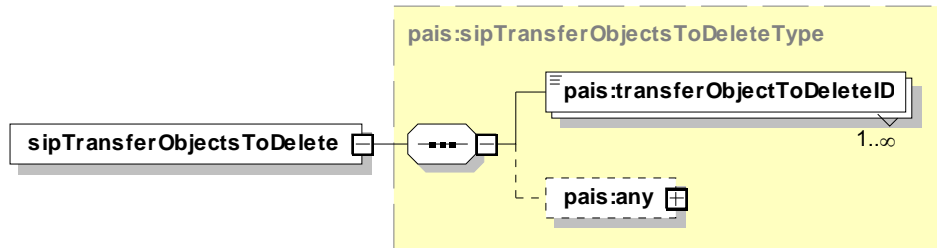


Figure 6-6: SIP Transfer Object To Delete

- user-defined attributes:
 - any.

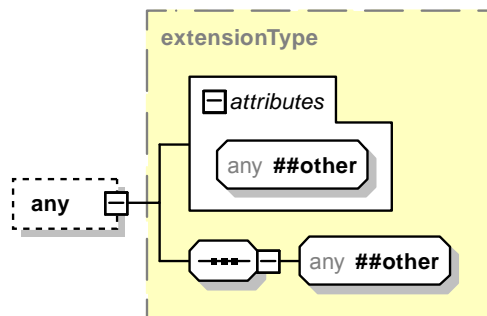


Figure 6-7: Any Extension Type

6.2.4 APPLYING SIP SCHEMAS TO XFDU

6.2.4.1 The SIP global information shall be inserted into the XFDU Package Header.

6.2.4.2 The SIP schema element <sipGlobalInformation> shall be mapped into the XFDU manifest element <environmentInfo> under <packageHeader> using the XFDU <extension> element as is shown schematically in figure 6-8.

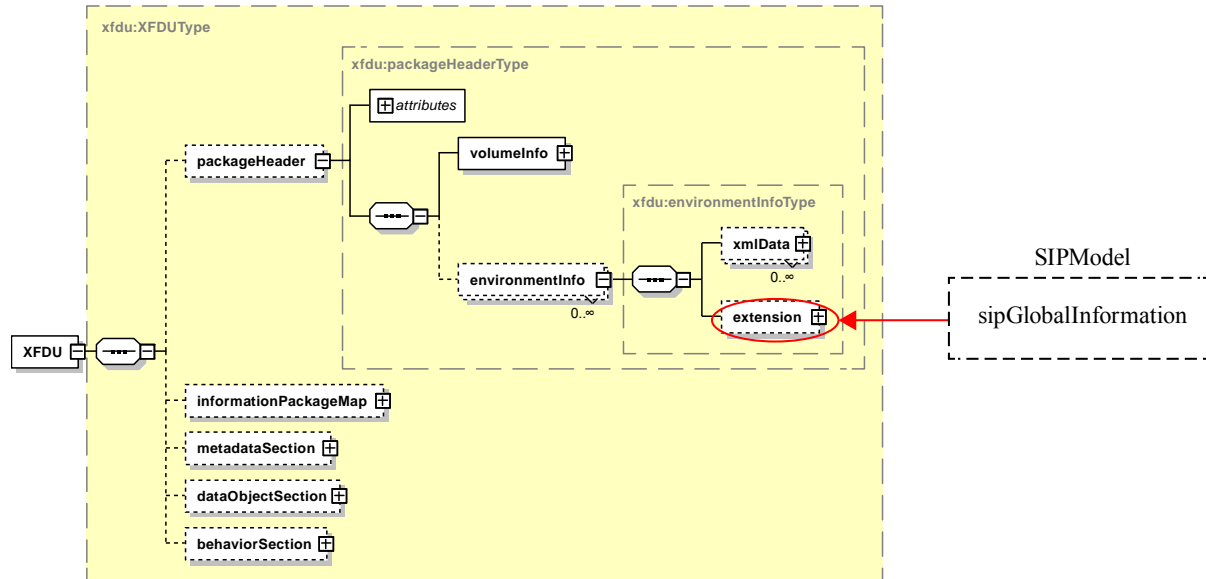


Figure 6-8: sipGlobalInformation Mapped to XFDU

6.2.4.3 SIP Transfer Objects, SIP Transfer Object Groups, SIP Data Objects, and SIP Transfer Objects to Delete shall be mapped into Content Units of the XFDU <informationPackageMap> using the <extension> element as shown in figure 6-9.

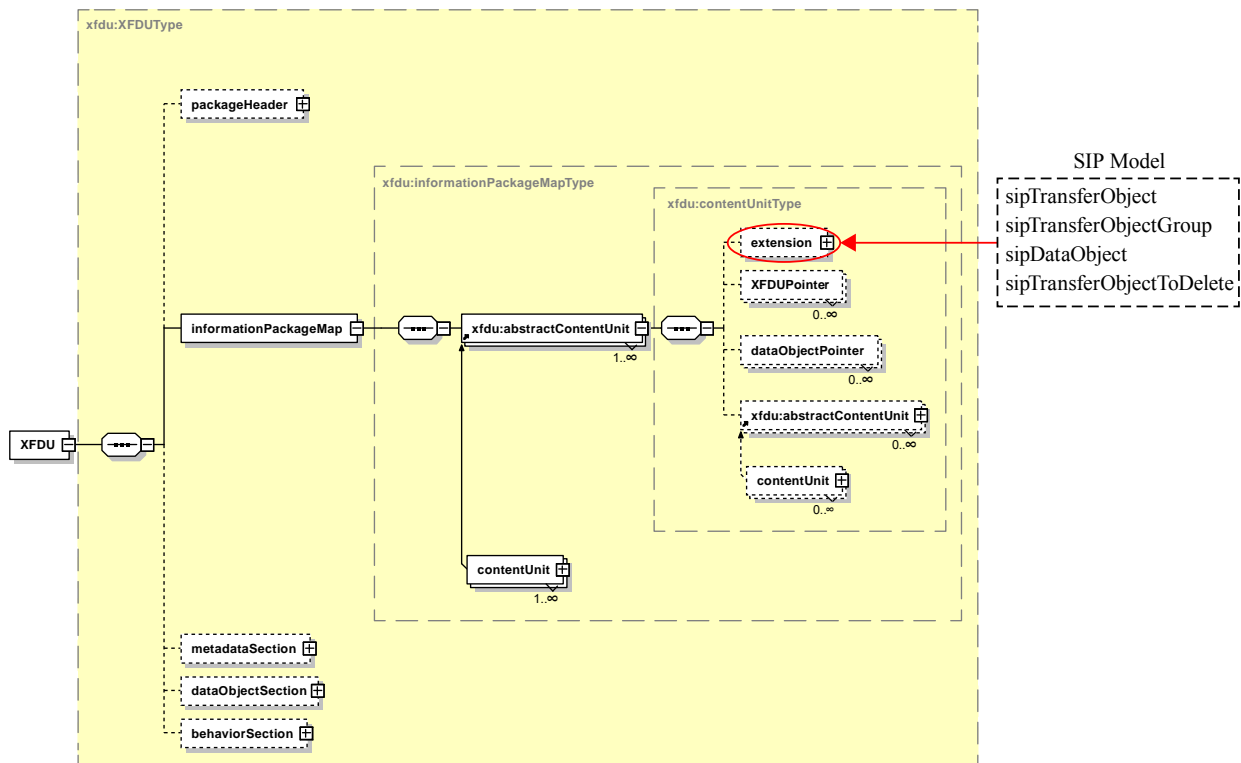


Figure 6-9: sipTransferObject Mapped to XFDU

6.2.4.4 A SIP Byte Stream shall be implemented under one <dataObject> in the <dataObjectSection> of the XFDU using the <byteStream> element as shown in figure 6-10.

NOTE – Also shown in figure 6-10 is the optional SIP Byte Stream Checksum implemented under the XFDU <byteStream> element using the XFDU <checksum> element.

6.2.4.5 A <dataObject> with its <byteStream> shall be referenced from the SIP Data Object incorporated as a Content Unit, as shown in figure 6-9, using the associated XFDU <dataObjectPointer>.

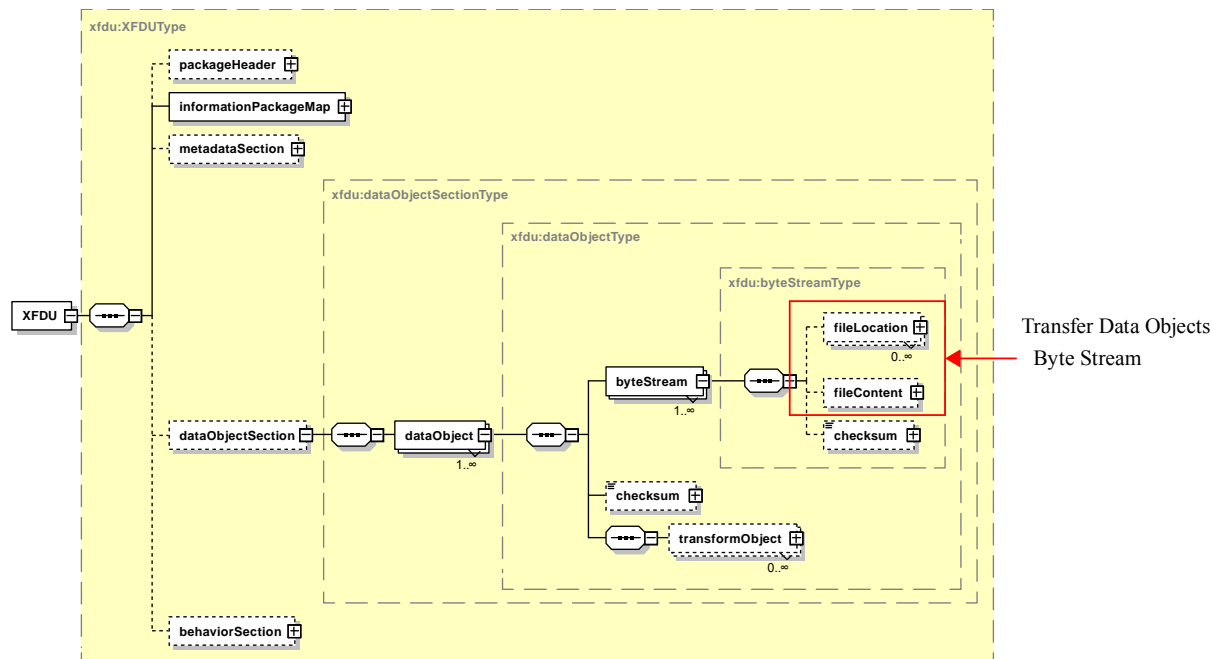


Figure 6-10: Byte Stream Mapped to XFDU byteStream Element

NOTE – As also shown in figure 6-10, the XFDU <byteStream> element allows any number of <fileLocation> elements and multiple <fileContent> elements.

6.2.4.6 For this Recommended Standard only a single <fileLocation> element and/or a single <fileContent> element shall be used.

NOTES

- 1 A SIP byte steam may be carried as a file in the XFDU file that is referenced by the <fileLocation> element. This same element may also reference a SIP byte stream that is external to the SIP XFDU. It is also possible to embed a SIP byte stream into the XFDU manifest file using the <fileContent> element. If both XFDU elements are used, the byte stream from the <fileContent> element is considered to be backup for the byte stream referenced by the <fileLocation> element as specified by XFDU semantics.
- 2 A detailed example of SIP to XFDU mappings is given in annex F.

6.3 SPECIALIZATION OF THE SIP IMPLEMENTATION IN AN XFDU

6.3.1.1 The SIP implementation in an XFDU defined in this document may be specialized to meet the needs of the Producer-Archive Project.

6.3.1.2 If SIP implementation specialization is required, it may only be specialized through the following modifications:

- addition of new elements (or sequence of elements) by the use of the element ‘any’;
- suppression of optional elements;
- optional elements turned to mandatory;
- modification of occurrence numbers that narrows the acceptable range (for example 10..15 instead of 1..n);
- definition of a list (enumeration) for strings (for example ‘groupStructureName’ =limited list of ‘set’, ‘sequence’, ‘directory’, ‘undescribed’);
- definition of patterns for identifiers (imposed nomenclature) or restriction of string length.

ANNEX A

PAIS XML SCHEMAS

(NORMATIVE)

A1 COMMON TYPES

The filename of this schema is 'ccsds-pais-common-types.xsd'.

If this name is changed, then the references to this schema from the other PAIS schemas must be updated.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pais="urn:ccsds:schema:pais:1"
targetNamespace="urn:ccsds:schema:pais:1" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xsd:complexType name="associationType">
    <xsd:annotation>
      <xsd:documentation>
        A relationship from this object type or collection to the object
        type or collection having the identifier given by the targetID element
        (targeted object types include Transfer Object Type, Transfer Object Group
        Type, and Data Object Type). These are transversal links specifying the
        type of relation.</xsd:documentation>
      </xsd:annotation>
      <xsd:sequence>
        <xsd:element name="targetID" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>
              Identifier of target object type or collection of the Producer-Archive
              Project to which the relationship is established. Possible target object
              types and their corresponding identifiers include: (Transfer Object Type,
              descriptorID), (Transfer Object Group Type, transferObjectGroupID), (Data
              Object Type, dataObjectTypeID), collection,
              descriptorID).</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="relationDescription" maxOccurs="unbounded">
            <xsd:annotation>
              <xsd:documentation>Description of the relationship or role played
              by this object type or collection with respect to the target object type or
              collection.</xsd:documentation>
            </xsd:annotation>
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="relationType" type="xsd:string">
                  <xsd:annotation>
                    <xsd:documentation>Name for the relation (e.g., Data Entity
                    Dictionary (DED)), Syntax, Context, Provenance, Reference, Fixity
                    Information) played by this object type or collection with respect to the
                    target object type or collection.</xsd:documentation>
                  </xsd:annotation>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:annotation>
  </xsd:complexType>
</xsd:schema>
```



```

        </xsd:annotation>
    </xsd:element>
    <xsd:element name="relationTextualDescription"
type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Text description of the
relation.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="extensionType">
    <xsd:annotation>
        <xsd:documentation>Allows third parties to define extensions from a
namespace controlled by the third party. The extension element allows one
single element that may be complex (and thus may incorporate additional
elements).</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:any namespace="##other" processContents="lax"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<xsd:complexType name="occurrenceType">
    <xsd:annotation>
        <xsd:documentation>The number of instances of this object type in
each instance of its containing group. This may be expressed as one, or
more, or as a range of values. This number may not be known at the time of
specification. In the case of a unique value, the Minimal Occurrence and
the Maximal Occurrence shall have the same value.</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="minOccurrence" type="xsd:nonNegativeInteger">
            <xsd:annotation>
                <xsd:documentation>Non negative Integer value (less than or equal
to maxOccurrence).</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:choice>
            <xsd:element name="maxOccurrence" type="xsd:nonNegativeInteger">
                <xsd:annotation>
                    <xsd:documentation>Non negative Integer value (greater than or
equal to minOccurrence).</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="maxUnknown" type="xsd:anySimpleType">
                <xsd:annotation>
                    <xsd:documentation>Empty element when used</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:choice>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

A2 TRANSFER OBJECT TYPE DESCRIPTOR MODEL

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pais="urn:ccsds:schema:pais:1"
targetNamespace="urn:ccsds:schema:pais:1" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xsd:include schemaLocation="ccsds-pais-common-types.xsd"/>
  <xsd:element name="transferObjectTypeDescriptor">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="identification">
          <xsd:annotation>
            <xsd:documentation>Parent element supporting the unique
identification of the Transfer Object Type Descriptor within the Producer-
Archive Project.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="descriptorModelID" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Identifier of the Transfer Object Type
Descriptor Model upon which this Descriptor is based. It may be the
Descriptor Model as given in the standard or it may be a specialized
version. It shall be unique across the Producer-Archive Project and shall
be provided by the Archive.
The standard value for this ID, when no specialization has been employed,
is 'CCSD0014'. When specialization has been employed, the value for the
Descriptor Model ID should be agreed between the Producer and the
Archive.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="descriptorModelVersion" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Identifier of the version of the
Descriptor Model. This allows tracking updates to the identified Descriptor
Model. The standard value for this ID, when the Descriptor Model ID is
'CCSD0014' and no revisions to this standard model have occurred, is
'V1.0'. When specialization has been employed, the value for the Descriptor
Model Version shall be agreed between the Producer and the
Archive.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="descriptorID" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>Identifier of this Descriptor within
the Producer-Archive Project. Uniqueness shall be checked by the Archive.
This identifier can be used by a relationship element to refer to this
Transfer Object Type.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="producerSourceID" type="xsd:string"
minOccurs="0">
          <xsd:annotation>
            <xsd:documentation> Identifier of the Producer Source
sufficient to enable the Archive to identify the origin of the SIP and to
send an acknowledgement. There may be multiple Producer Sources sending
SIPs to the Archive within a given Producer-Archive Project. The form of
the Producer Source ID needs agreement between Producer and Archive. If

```

specified, only those Producer Sources identified here may include Transfer Objects, of this Transfer Object Type, in SIPs.</xsd:documentation>

```

    </xsd:annotation>
  </xsd:element>
  <xsd:element name="any" type="pais:extensionType"
minOccurs="0">
    <xsd:annotation>
      <xsd:documentation> Allows a conforming XML document to
have any additional XML elements at this point in its
structure.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="description">
  <xsd:annotation>
    <xsd:documentation> Parent element giving a high level
description of the Transfer Object Type.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="transferObjectTypeTitle"
type="xsd:string">
        <xsd:annotation>
          <xsd:documentation> Extensive descriptive phrase used as
the name of the Transfer Object Type.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="transferObjectTypeDescription"
type="xsd:string">
        <xsd:annotation>
          <xsd:documentation> Explanatory text describing the
meaning of the Transfer Object Type.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="transferObjectTypeOccurrence"
type="pais:occurrenceType">
        <xsd:annotation>
          <xsd:documentation>Number of Transfer Object Type
instances (Transfer Objects) of this Transfer Object Type within the
Producer-Archive Project. This may be expressed as one, or more, or as a
range of values. This number may not be known at the time of descriptor
creation. In the case of a unique value, minOccurs = maxOccurrence =
value.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="transferObjectTypeSize" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Characterization of the size range for
Transfer Objects of this type. The size range may not be known at the time
of descriptor creation and therefore the corresponding element may be
omitted.</xsd:documentation>
        </xsd:annotation>
      </xsd:complexType>
      <xsd:sequence>
        <xsd:element name="minSize" type="xsd:float"
minOccurs="0"/>
        <xsd:element name="maxSize" type="xsd:float"
minOccurs="0"/>

```

```

        <xsd:element name="unitsType" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="KB"/>
                    <xsd:enumeration value="MB"/>
                    <xsd:enumeration value="GB"/>
                    <xsd:enumeration value="TB"/>
                    <xsd:enumeration value="PB"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="namePreservationRule" type="xsd:string"
minOccurs="0">
    <xsd:annotation>
        <xsd:documentation> Specification from the Producer
describing how to recognize and/or create the preservable name to be given
to each Data Object. It shall apply to all the Data Objects under this
Transfer Object Type. It may assist the Producer in creating the SIPs and
it may assist the Archive in preserving appropriate
information.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="any" type="pais:extensionType"
minOccurs="0">
    <xsd:annotation>
        <xsd:documentation>Allows a conforming XML document to
have any additional XML elements at this point in its
structure.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="relation">
    <xsd:annotation>
        <xsd:documentation>Parent element that identifies the
collection to which the Transfer Object Types under this Descriptor are a
part, and optionally that specify directional relationships between these
Transfer Object Types and other Data Object Types, Transfer Object Types,
Transfer Object Group Types and collections. These relationships are
specified using identifiers as given within the Descriptors of the
MOT.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="parentCollection" type="xsd:string">
                <xsd:annotation>
                    <xsd:documentation>Identifier of a Collection Descriptor
that provides an aggregation view that includes this Transfer Object Type.
The highest level Descriptor in the MOT shall be a Collection
Descriptor.</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="association" type="pais:associationType"
minOccurs="0" maxOccurs="unbounded">
                <xsd:annotation>

```

```

        <xsd:documentation>Parent element used to describe a
relationship from this Transfer Object Type under this Descriptor to the
object type or collection having the identifier given by the targetID
element. (targeted object types include Transfer Object Type, Transfer
Object Group Type, and Data Object Type). These are transversal links
specifying the type of relation.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="any" type="pais:extensionType"
minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Allows a conforming XML document to
have any additional XML elements at this point in its
structure.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="groupType" type="pais:transferObjectGroupType"
maxOccurs="unbounded">
    <xsd:annotation>
        <xsd:documentation>Parent element that identifies one or
several Group Types. A Group Type is a set of characteristics describing
zero or more Data Object Types, even complex Data Objects Types made up of
several parts. The Transfer Object Group Type may contain additional
Transfer Object Group Types in support of complex structures such as
directory structures.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="any" type="pais:extensionType" minOccurs="0">
    <xsd:annotation>
        <xsd:documentation>Allows a conforming XML document to have any
additional XML elements at this point in its structure.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="dataObjectType">
    <xsd:sequence>
        <xsd:element name="dataObjectTypeID" type="xsd:string">
            <xsd:annotation>
                <xsd:documentation>Identifier of the Data Object Type. It shall
be unique across the Producer-Archive Project. Uniqueness shall be checked
by the Archive. This identifier can be used by a relationship element to
refer to this Data Object Type.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="dataObjectTypeDescription" type="xsd:string"
minOccurs="0">
            <xsd:annotation>
                <xsd:documentation>Explanatory text describing the meaning of
this Data Object Type.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="dataObjectTypeOccurrence"
type="pais:occurrenceType">
            <xsd:annotation>

```

`<xsd:documentation>`Parent element supporting the number of instances (Data Objects) of this Data Object Type in each instance of its containing group. This may be expressed as one, or more, or as a range of values. This number may not be known at the time of descriptor creation and instantiation. In the case of a unique value, then `minOccurrence = maxOccurrence = value.``</xsd:documentation>`

`</xsd:annotation>`

`</xsd:element>`

`<xsd:element name="dataObjectTypeFileOccurrence" type="pais:occurrenceType" minOccurs="0">`

`<xsd:annotation>`

`<xsd:documentation>`Parent element supporting the number of file instances in each Data Object Type instance. This may be expressed as one, or more, or as a range of values. This number may not be known at the time of descriptor creation and instantiation. In the case of a unique value, then `minOccurrence = maxOccurrence = value.` The absence of this element implies there shall be one file instance.`</xsd:documentation>`

`</xsd:annotation>`

`</xsd:element>`

`<xsd:element name="dataObjectTypeFormat" minOccurs="0">`

`<xsd:annotation>`

`<xsd:documentation>`Parent element supporting the identification of the format of this Data Object Type in the Transfer Object Group Type using a MIME type or a registration mechanism.`</xsd:documentation>`

`</xsd:annotation>`

`<xsd:complexType>`

`<xsd:sequence>`

`<xsd:element name="mimeType" type="xsd:string" minOccurs="0">`

`<xsd:annotation>`

`<xsd:documentation>`MIME type for the Data Object Type (e.g., "application/octet-stream" MIME type as documented by the Internet Engineering Task Force (IETF)).`</xsd:documentation>`

`</xsd:annotation>`

`</xsd:element>`

`<xsd:element name="registrationInformation" minOccurs="0">`

`<xsd:annotation>`

`<xsd:documentation>`Parent element supporting information used to identify the format of this Data Object Type as registered with a specified registration authority.`</xsd:documentation>`

`</xsd:annotation>`

`<xsd:complexType>`

`<xsd:sequence>`

`<xsd:element name="registrationAuthority" type="xsd:string" minOccurs="0">`

`<xsd:annotation>`

`<xsd:documentation>`Identifier of the organization or system that has registered and holds the format description (e.g., IETF, CCSDS Control Authority(CA)).`</xsd:documentation>`

`</xsd:annotation>`

`</xsd:element>`

`<xsd:element name="registeredID" type="xsd:string" minOccurs="0">`

`<xsd:annotation>`

`<xsd:documentation>`Identifier of the format description within the context of the registration authority (e.g., NSSD0145 under CCSDS CA).`</xsd:documentation>`

`</xsd:annotation>`

`</xsd:element>`

`</xsd:sequence>`

`</xsd:complexType>`

```

        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="dataObjectTypeEncoded" type="pais:encodingType"
minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>Characterization of how a Data Object has been
encoded. May be used to describe possible nested encodings, for example a
Data Object encoded in a zip file and then encoded in a binhex
file.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="dataObjectTypeAssociation"
type="pais:associationType" minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>Parent element supporting the description of a
relationship from this Data Object Type to the object type or collection
having the identifier given by the targetID element (targeted object types
include Data Object Type, Transfer Object Type, and Transfer Object Group
Type). These are transversal links specifying the type of
relation.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="any" type="pais:extensionType" minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>Allows a conforming XML document to have any
additional XML elements at this point in its structure.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="transferObjectGroupType">
  <xsd:sequence>
    <xsd:element name="groupTypeID" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>Identifier of a Transfer Object Group Type
that shall be used to distinguish this Group Type from other Group Types.
It shall be unique within a given Producer-Archive Project. Uniqueness
shall be checked by the Archive. This identifier may be used by a
relationship element to refer to this Transfer Object Group
Type.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="groupTypeDescription" type="xsd:string"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>Explanatory text describing the meaning of
this Transfer Object Group Type.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="groupTypeStructureName" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>A name for the type of Transfer Object Group
Type, such as directory, set, sequence, undescribed, or other name not
included in this list.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="groupTypeEncoded" type="pais:encodingType"
minOccurs="0" maxOccurs="unbounded">

```

```

    <xsd:annotation>
      <xsd:documentation>Characterisation of how groups of this type
are encoded. It may be used iteratively to describe possible nested
encodings and in which case they are described in the order of application.
The result of encoding shall be a single Data Object. The possible presence
of Data Object Type and any additional Transfer Object Group Type
specifications under this Transfer Object Group Type specification shall be
understood to comprise a detailed model of this group after the encoding
has been reversed. Validation of the encoded content may not be possible
because the resulting data structure will not carry any of the identifiers
associated with the detailed modeling of this group.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="groupTypeOccurrence" type="pais:occurrenceType"
minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>Parent element supporting the number of
instances (Transfer Object Groups) of this Transfer Object Group Type in
each instance of its containing group or, if there is no containing group,
in each containing Transfer Object Type instance. This may be expressed as
one, or more, or as a range of values. This number may not be known at the
time of descriptor creation. In the case of a unique value, then
minOccurrence = maxOccurrence = value.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="groupTypeAssociation" type="pais:associationType"
minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>Parent element used to describe a relationship
from this Transfer Object Group Type to the object type or collection
having the identifier given by the targetID element (targeted object types
include Transfer Object Type, Transfer Object Group Type, and Data Object
Type). These are transversal links specifying the type of
relation.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="dataObjectType" type="pais:dataObjectType"
minOccurs="0" maxOccurs="unbounded">
    <xsd:annotation>
      <xsd:documentation>Parent element supporting a description of the
Data Object Type. A Data Object Type describes one or more Data Objects
(instances). There may be multiple Data Object Types within a Transfer
Object Group Type.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="groupType" type="pais:transferObjectGroupType"
minOccurs="0" maxOccurs="unbounded"/>
  <xsd:element name="any" type="pais:extensionType" minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>Allows a conforming XML document to have any
additional XML elements at this point in its structure.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="encodingType">
  <xsd:sequence>
    <xsd:element name="encodingName" type="xsd:string">
      <xsd:annotation>

```


RECOMMENDED STANDARD FOR PRODUCER-ARCHIVE INTERFACE SPECIFICATION

```

    <xsd:documentation>A name for the type of encoding, such as zip,
tar, or other name not included in this list.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="encodingDescription" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>
      Description or reference to the type of the
encoding.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

A3 COLLECTION DESCRIPTOR MODEL

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pais="urn:ccsds:schema:pais:1"
targetNamespace="urn:ccsds:schema:pais:1" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xsd:include schemaLocation="ccsds-pais-common-types.xsd"/>
  <xsd:element name="collectionDescriptor">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="identification">
          <xsd:annotation>
            <xsd:documentation>
              Parent element supporting the unique identification of a
              collection within the Producer-Archive Project.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="descriptorModelID" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>Identifier of the Descriptor Model
                upon which this Collection Descriptor is based. It may be the Descriptor
                Model as given in the standard or it may be a specialized version. It shall
                be unique across the Producer-Archive Project and shall be provided by the
                Archive. The standard value for this ID, when no specialization has been
                employed, is 'CCSD0015'. When specialization has been employed, the value
                for the Descriptor Model ID should be agreed between the Producer and the
                Archive.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="descriptorModelVersion" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>Identifier of the version of the
                Collection Descriptor Model. This allows tracking updates to the identified
                Descriptor Model. The standard value for this ID, when the Descriptor Model
                ID is 'CCSD0015' and no revision to this standard model has occurred, is
                'V1.0'. When specialization has been employed, the value for the Descriptor
                Model Version shall be agreed between the Producer and the
                Archive.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="descriptorID" type="xsd:string">
              <xsd:annotation>
                <xsd:documentation>Identifier of this Descriptor within
                the Producer-Archive Project. Uniqueness shall be checked by the Archive.
                This identifier may be used by a relationship element to refer to this
                collection. The highest-level Descriptor in the MOT shall be a Collection
                Descriptor. The value for its Descriptor ID shall be the Producer-Archive
                Project ID. The Producer-Archive Project ID shall be assigned by the
                Archive.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="any" type="pais:extensionType"
minOccurs="0">
              <xsd:annotation>

```

```

        <xsd:documentation>Allows a conforming XML document to
have any additional XML elements at this point in its
structure.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="description">
    <xsd:annotation>
        <xsd:documentation>Parent element giving a high level
description of the collection.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="collectionTitle" type="xsd:string">
                <xsd:annotation>
                    <xsd:documentation>Extensive descriptive phrase used as
the name of the collection.</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="collectionDescription" type="xsd:string">
                <xsd:annotation>
                    <xsd:documentation>Explanatory text describing the
meaning of the collection.</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:element name="collectionSize" minOccurs="0">
                <xsd:annotation>
                    <xsd:documentation>Characterization of the size range of
this collection. The size range may not be known at the time of descriptor
creation and therefore the corresponding element may be
omitted.</xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="minSize" type="xsd:float"
minOccurs="0"/>
                        <xsd:element name="maxSize" type="xsd:float"
minOccurs="0"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="unitsType" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="KB"/>
                        <xsd:enumeration value="MB"/>
                        <xsd:enumeration value="GB"/>
                        <xsd:enumeration value="TB"/>
                        <xsd:enumeration value="PB"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
    <xsd:element name="any" type="pais:extensionType"
minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>Allows a conforming XML document to
have any additional XML elements at this point in its
structure.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>

```

```

        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="relation">
  <xsd:annotation>
    <xsd:documentation>Parent element identifying directional
relationships between this collection and other collections, Transfer
Object Types, Transfer Object Group Types, or Data Object Types. These
relationships are specified using identifiers as given within the
Descriptors of the MOT.</xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="parentCollection" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>Identifier of a Collection Descriptor
that provides an aggregation view that includes this collection. The value
for the parentCollection of the highest level Descriptor in the MOT shall
be 'none'.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="association" type="pais:associationType"
minOccurs="0" maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>Parent element used to describe a
relationship from this collection to the object type or collection having
the identifier given by the targetID element (targeted object type include
Transfer Object Type, Transfer Object Group Type, and Data Object Type).
These are transversal links specifying the type of
relation.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="any" type="pais:extensionType"
minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Allows a conforming XML document to
have any additional XML elements at this point in its
structure.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="any" type="pais:extensionType" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>Allows a conforming XML document to have any
additional XML elements at this point in its structure.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

A4 SIP CONSTRAINTS XML SCHEMA

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:pais="urn:ccsds:schema:pais:1"
targetNamespace="urn:ccsds:schema:pais:1" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xsd:include schemaLocation="ccsds-pais-common-types.xsd"/>
  <xsd:element name="sipConstraints">
    <xsd:annotation>
      <xsd:documentation>Two types of constraints apply to SIPs: the first
one defines the authorized content of each SIP; the second one defines the
order in which SIPs have to be delivered by the Producer to the Archive.
The constraints are associated with a specific Producer-Archive
Project.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="producerArchiveProjectID" type="xsd:string">
          <xsd:annotation>
            <xsd:documentation>A unique identifier of the Producer-Archive
Project among all Producer-Archive Projects undertaken by this Archive.
This ID shall be provided by the Archive for use in the
SIPs.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="sipContentType" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation>A set of elements describing the content of
each type of SIP.</xsd:documentation>
          </xsd:annotation>
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="sipContentTypeID" type="xsd:string">
                <xsd:annotation>
                  <xsd:documentation>Identifier of the SIP Content Type. It
shall be unique across the Producer-Archive Project. Uniqueness shall be
checked by the Archive. It shall be referenced in each created
SIP.</xsd:documentation>
                </xsd:annotation>
              </xsd:element>
              <xsd:element name="authorizedDescriptor"
maxOccurs="unbounded">
                <xsd:annotation>
                  <xsd:documentation>A list of the Transfer Object Type
Descriptor Identifiers that a transferred SIP of the type "SIP Content Type
ID", may or may not contain. A transferred SIP shall not contain a Transfer
Object associated with a Descriptor ID that is not in this
list.</xsd:documentation>
                </xsd:annotation>
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="descriptorID" type="xsd:string">
                      <xsd:annotation>
                        <xsd:documentation>Identifier for this Transfer
Object Type which is allowed in this type of SIP.</xsd:documentation>
                      </xsd:annotation>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

```

```

        <xsd:element name="occurrence"
type="pais:occurrenceType">
        <xsd:annotation>
        <xsd:documentation>Number of instances (Transfer
Objects) of this Transfer Object Type that will be present in a SIP of this
type. This may be expressed as one, or more, or as a range of values. If
the number of Transfer Objects is not known at the time of SIP Content Type
specification creation, then zero shall be specified as the minOccurrence
and the MaxUnknown element shall be used. In the case of a unique value,
the minOccurrence and the maxOccurrence shall have the same
value.</xsd:documentation>
        </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="sipSequencingConstraintGroup" minOccurs="0"
maxOccurs="unbounded">
    <xsd:annotation>
    <xsd:documentation>A set of elements describing the delivery
order applying to the SIP Content Types within the Producer-Archive
Project. There are no constraints between the SIPs belonging to different
SIP Sequencing Constraint Groups.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element name="groupName" type="xsd:string"
minOccurs="0">
        <xsd:annotation>
        <xsd:documentation>Name for this group of constraint
items.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="constraintItem" minOccurs="2"
maxOccurs="unbounded">
        <xsd:complexType>
        <xsd:sequence>
        <xsd:element name="sipContentTypeID" type="xsd:string">
            <xsd:annotation>
            <xsd:documentation>Identifier of a SIP Content
Type. It identifies each of the types of SIPs that are under constraints in
the corresponding group.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="constraintSerialNumber"
type="xsd:integer">
            <xsd:annotation>
            <xsd:documentation>A number that specifies the
order of delivery of SIPs with the associated SIP Content Type identifier
All SIPs of a SIP Content Type associated with a Constraint Serial Number
with a value of "n" must be delivered before all SIPs of a SIP Content Type
associated with a Constraint Serial Number with a value greater than "n".
There are no constraints between two or more SIP Content Types with the
same Constraint Serial Number value inside the same
group.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
    </xsd:complexType>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

```

```
        </xsd:sequence>  
      </xsd:complexType>  
    </xsd:element>  
  </xsd:sequence>  
 </xsd:complexType>  
 </xsd:element>  
 </xsd:sequence>  
 </xsd:complexType>  
 </xsd:element>  
 </xsd:schema>
```

A5 SIP MODEL XML SCHEMA

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:pais="urn:ccsds:schema:pais:1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:ccsds:schema:pais:1" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xsd:include schemaLocation="ccsds-pais-common-types.xsd"/>
  <xsd:element name="sipGlobalInformation"
type="pais:sipGlobalInformationType">
    <xsd:annotation>
      <xsd:documentation>Parent element supporting the unique
identification of each SIP within the Producer-Archive Project and the
ability to optionally track the sequencing of SIPs.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="sipTransferObject" type="pais:sipTransferObjectType">
    <xsd:annotation>
      <xsd:documentation>Parent element that conceptually holds two types
of containers. The Transfer Object container consists of: one Transfer
Object Identification and Status container; and one or more Transfer Object
Group containers.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="sipTransferObjectGroup"
type="pais:sipTransferObjectGroupType">
    <xsd:annotation>
      <xsd:documentation>Conceptually holds three types of containers.
These are the Transfer Object Group Identification container; any number of
other Transfer Object Group containers; and any number of Data Object
containers.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="sipTransferObjectsToDelete"
type="pais:sipTransferObjectsToDeleteType">
    <xsd:annotation>
      <xsd:documentation>Composed of one or more elements giving the
identification of the Transfer Objects previously sent to the Archive that
must be deleted by the Archive.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="sipDataObject" type="pais:sipDataObjectType">
    <xsd:annotation>
      <xsd:documentation>Conceptually holds two or more containers. These
are the: Data Object Identification container; and one or more Byte Stream
containers.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:complexType name="sipGlobalInformationType">
    <xsd:sequence>
      <xsd:element name="sipID" type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>Identifier of the delivered SIP within the
context of the given Producer-Archive Project. If there are multiple
Producer Sources submitting SIPs within a single Producer-Archive
Project, this SIP ID must be unique across all such Producer Sources. It is
inserted during SIP construction. The form shall be agreed between Producer
and Archive, but the identifier shall be generated by the Producer. The
Archive shall check uniqueness.</xsd:documentation>

```



```

    </xsd:annotation>
  </xsd:element>
  <xsd:element name="producerSourceID" type="xsd:string">
    <xsd:annotation>
      <xsd:documentation>Identifier of the Producer Source sufficient
to enable the Archive to identify the origin of the SIP and to send an
acknowledgement. There may be multiple Producer sources sending SIPs to the
Archive within a given Producer- Archive Project. The form of the Producer
Source ID shall be agreed between Producer and Archive.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="producerArchiveProjectID" type="xsd:string">
    <xsd:annotation>
      <xsd:documentation>Identifier of the Producer-Archive Project
that distinguishes the project from all other Producer-Archive Projects
undertaken by this Archive. This ID shall be provided by the Archive for
use in the SIPs.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="sipContentTypeID" type="xsd:string">
    <xsd:annotation>
      <xsd:documentation>Identifier of the specification that defines
which Transfer Object Types (i.e., Descriptor IDs) are allowed within this
SIP, as well as their occurrence within the SIP.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="sipSequenceNumber" type="xsd:integer"
minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>A number indicating the order in which the SIP
has been sent. This number is unique within the combined context of the
Producer-Archive Project and Producer Source ID. This becomes mandatory for
all SIPs sent by a Producer Source if any of the Transfer Objects to be
provided by the Producer Source have a Descriptor that does not specify a
unique value for the number of Transfer Objects to be delivered. It shall
be generated by the Producer.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element name="any" type="pais:extensionType" minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>Allows a conforming XML document to have any
additional XML elements at this point in its structure.</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="sipTransferObjectType">
  <xsd:sequence>
    <xsd:element name="descriptorID" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>An identifier of the Transfer Object Type
Descriptor that describes this type of Transfer Object. This is obtained
from the MOT.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="transferObjectID" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>A unique identifier for each delivered
Transfer Object within the Producer-Archive Project. It is inserted during
SIP building. The form shall be agreed between Producer and Archive. It

```

shall be generated by the Producer. Uniqueness shall be checked by the Archive.

```

</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="lastTransferObjectFlag" minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>An indicator specifying that this Transfer
Object is the last Transfer Object of this type (i.e., within the scope of
this Descriptor) being delivered by this Producer Source. This element is
particularly useful when the number of Transfer Objects to be delivered is
not known in advance. If used with a single Producer Source for Transfer
Objects of this type, this flag eliminates the need for an additional
contact between the Archives and the Producer Source to verify that all
such Transfer Objects have been received. If there are multiple Producer
Sources that may be delivering Transfer Objects of this type, the Archive
may or may not need to contact these Producer Sources to determine when all
such Transfer Objects have been sent and received.</xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="TRUE"/>
      <xsd:enumeration value="FALSE"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="replacementTransferObjectID" type="xsd:string"
minOccurs="0">
  <xsd:annotation>
    <xsd:documentation>An identifier of the Transfer Object ID of a
previously sent Transfer Object that is to be replaced by this Transfer
Object.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="any" type="pais:extensionType" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="sipTransferObjectGroupType">
  <xsd:sequence>
    <xsd:element name="associatedDescriptorGroupTypeID"
type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>The identifier of the associated group
description within the associated Descriptor. If this group is an instance
of a Transfer Object Group Type as specified in the Descriptor, then the
value of this element shall be the Transfer Object Group Type ID of that
Transfer Object Group Type. If this group is an instance that is part of a
data structure transferred under a Transfer Object Group Type whose
Transfer Object Group Structure Name has the value 'undescribed', then
the value of this Associated Descriptor ID shall be the Transfer Object
Group Type ID of that Transfer Object Group Type.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:choice minOccurs="0">
      <xsd:element name="transferObjectGroupName"
type="xsd:string">
        <xsd:annotation>
          <xsd:documentation>A name given to the group, such as a
directory name, that is associated with the Transfer Object Group instance.
It shall be provided by the Producer. If the group has been modeled as a

```

```

directory (i.e., Structure Name = 'directory'), it is the name of the
directory excluding any path information.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="transferObjectGroupPreservationName"
type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>A name given to the group, such as a
directory name, that is to be preserved by the archive in association with
the Transfer Object Group instance. If the group has been modeled as a
directory (i.e., Structure Name = 'directory'), it is the name of the
directory excluding any path information.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:choice>
<xsd:element name="any" type="pais:extensionType" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="sipDataObjectType">
  <xsd:sequence>
    <xsd:element name="associatedDescriptorDataID" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>The identifier of the associated data
description within the associated Descriptor. If this is an instance of a
Data Object Type defined in the Descriptor then this shall be the Data
Object Type ID of that Data Object Type. If this is an instance of a
Transfer Object Group Type defined in the Descriptor to be encoded and thus
it results in a single Data Object, then this shall be the Transfer Object
Group Type ID of that Transfer Object Group Type. If this is an instance of
a Data Object that is transferred within the context of a Descriptor
defined Transfer Object Group Type whose Transfer Object Group Type
Structure Name has the value 'undescribed', then this shall be the Transfer
Object Group Type ID of that Transfer Object Group
Type.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="dataObjectPreservationName" type="xsd:string"
minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>A name to be preserved in association with the
Data Object instance. When the Data Object is composed of a single byte
stream, it tells the Archive exactly what name is to be preserved in
association with that byte stream. When the Data Object is composed of
multiple byte streams, it tells the Archive exactly what name is to be
preserved in association with the set of byte streams.It shall be provided
by the Producer.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="any" type="pais:extensionType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="sipTransferObjectsToDeleteType">
  <xsd:sequence>
    <xsd:element name="transferObjectToDeleteID" type="xsd:string"
maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>An identifier of the Transfer Object ID of a
previously sent Transfer Object that is to be deleted by the
Archive.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>

```

RECOMMENDED STANDARD FOR PRODUCER-ARCHIVE INTERFACE SPECIFICATION

```
</xsd:element>  
  <xsd:element name="any" type="pais:extensionType" minOccurs="0"/>  
</xsd:sequence>  
</xsd:complexType>  
</xsd:schema>
```

ANNEX B

LEGEND FOR XML FIGURES

(INFORMATIVE)

Element names and type names conform to the following naming conventions:

- Element names have all lowercase letters for the first word, with only the first letter uppercase in each subsequent word that appears in the name.
- Type names begin with a lowercase letter and end with the word ‘Type’.

All the XML schemas included in this document have been generated with the tool XML Spy. Thus the notations are those of XML Spy.

The following figure describes all these notations.

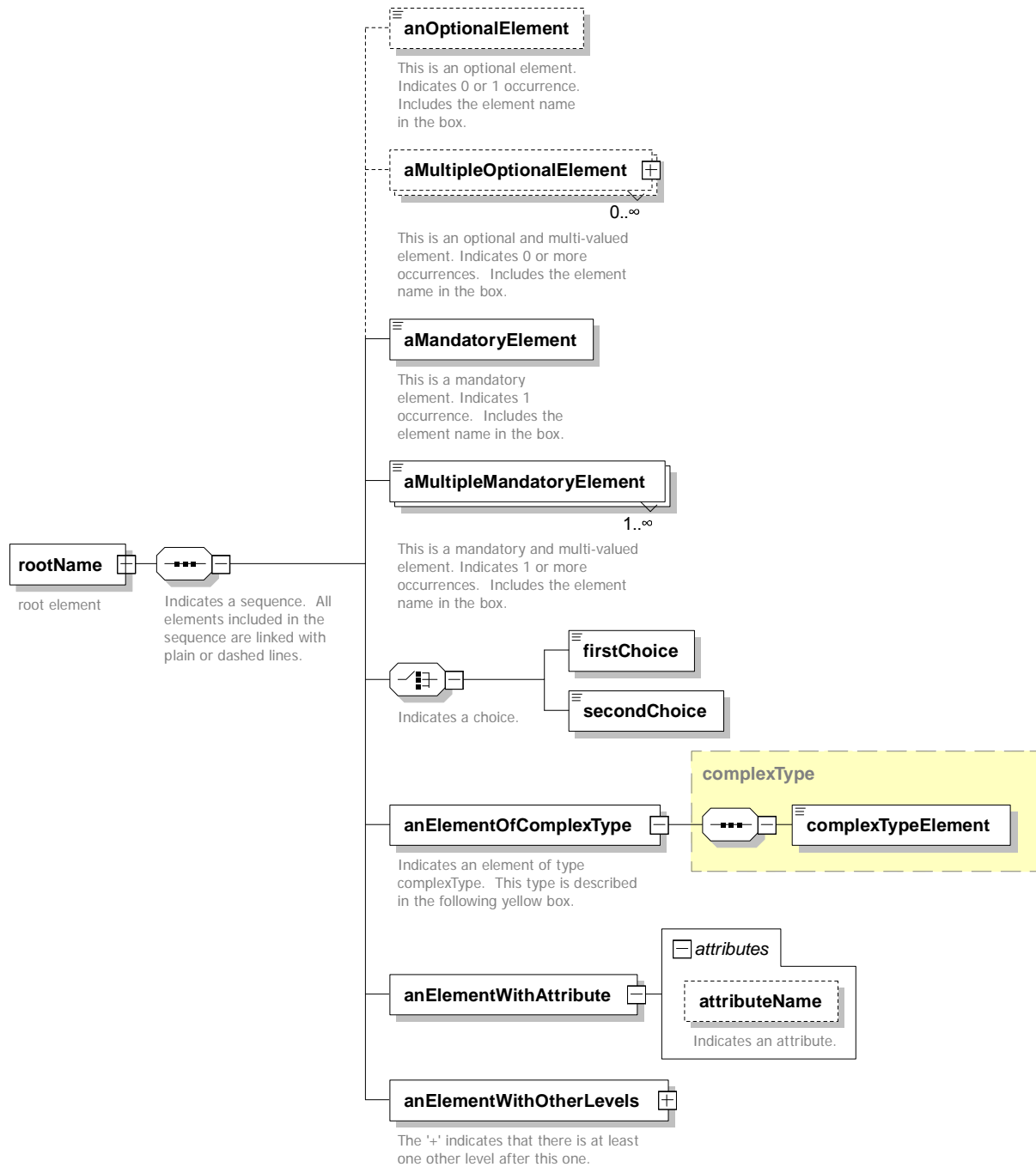


Figure B-1: Legend for XML Figures

ANNEX C

INFORMATIVE REFERENCES

(INFORMATIVE)

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

- [C1] *Organization and Processes for the Consultative Committee for Space Data Systems*. Issue 3. CCSDS Record (Yellow Book), CCSDS A02.1-Y-3. Washington, D.C.: CCSDS, July 2011.
- [C2] *Data Entity Dictionary Specification Language (DEDSL)—Abstract Syntax (CCSD0011)*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 647.1-B-1. Washington, D.C.: CCSDS, June 2001. [Equivalent to ISO 21961:2003.]
- [C3] *Data Entity Dictionary Specification Language (DEDSL)—XML/DTD Syntax (CCSD0013)*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 647.3-B-1. Washington, D.C.: CCSDS, January 2002. [Equivalent to ISO 22643:2003.]
- [C4] *The Data Description Language EAST Specification (CCSD0010)*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 644.0-B-3. Washington, D.C.: CCSDS, June 2010. [Equivalent to ISO 15889:2011.]
- [C5] *FGDC Standards Reference Model*. Reston, Virginia: Federal Geographic Data Committee, March 1996.
- [C6] Tim Bray, et al., eds. *Extensible Markup Language (XML) 1.0*. 5th ed. W3C Recommendation. N.p.: W3C, November 2008.
- [C7] Henry S. Thompson, et al., eds. *XML Schema Part 1: Structures*. 2nd ed. W3C Recommendation. N.p.: W3C, October 2004.
- [C8] Paul V. Biron and Ashok Malhotra, eds. *XML Schema Part 2: Datatypes*. 2nd ed. W3C Recommendation. N.p.: W3C, October 2004.

ANNEX D**MANAGEMENT OF IDS****(INFORMATIVE)**

This annex gathers information for all encountered IDs, either Descriptor or SIP IDs, as background material for negotiation between the Producer and the Archive in the management of these IDs for using this Recommended Standard.

ID	Who provides ID	Who checks uniqueness	PAIS section
DESCRIPTOR			
Descriptor Model ID	Archive	Archive	3.2.2.2 p 3-5 (and collection)
Descriptor ID	Decided between Producer and Archive	Archive	3.2.2.2 p 3-5 (and collection)
Transfer Object Group Type ID	Decided between Producer and Archive	Archive	3.2.2.5 p 3-9
Data Object Type ID	Decided between Producer and Archive	Archive	3.2.2.5 p 3-9
SIP			
Producer-Archive Project ID	Archive	Archive	3 p 3-1; 5 p 5-1
Producer Source ID	Decided between Producer and Archive	Archive	5 p 5-1
SIP Content Type ID	Decided between Producer and Archive	Archive	4.2 p 4-1
SIP ID	Producer	Archive	5 p 5-1
Transfer Object ID	Producer	Archive	5 p 5-1

ANNEX E

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

E1 SECURITY CONSIDERATIONS

E1.1 OVERVIEW

The CCSDS requires that there to be an informative annex which points out security considerations for implementations of its standards. This annex presents the results of an analysis of security considerations applied to the technologies specified in this Recommended Standard.

General guidance on security issues may be found in the informational (Green) Book CCSDS 350.0-G-2: *The Application of CCSDS Protocols to Secure Systems* and references therein.

Security services may need to be applied during the storage and transfer of the model information and to the transfer and storage of Submission Information Packages (SIPs). However, the scope of this specification is specifically the content of the model information and the content of the SIP. Therefore, the specification of such security services is outside the scope of this document.

If there is a reason to believe that non-authorized entities might be able to view or obtain the model information or SIP data, and if there is a need to ensure that non-authorized entities not be able to view or obtain the data, then confidentiality mechanisms need to be applied. If there is a need to ensure that the SIP data has not been modified in transit, then integrity mechanisms need to be applied.

Authenticity of the source of the SIP data is required, and thus authentication mechanisms need to be applied. It is possible that any combination of these three security services may need to be applied to ensure that the data is not disclosed, not altered, and remains authentic.

E1.2 SECURITY CONCERNS RELATED TO THIS RECOMMENDED STANDARD

E1.2.1 DATA PRIVACY

The XFDU SIP Implementation allows the use of the XFDU standard data structures to describe the encryption of any or all of the data objects contained within an XFDU SIP package.

E1.2.2 DATA INTEGRITY

The XFDU SIP Implementation allows the use of the XFDU standard data structure to specify the checksums of any or all of the data contained within an XFDU package.

E1.2.3 AUTHENTICATION OF COMMUNICATING ENTITIES

Authentication of communicating entities involved in the transport of data that complies with the specifications of this Recommended Standard should be provided by the systems on which this Recommended Standard is implemented. The consequences of failing to properly authenticate the communicating entities involved in the transport of data that complies with the specifications of this Recommended Standard could include potential loss, corruption, and theft of data and loss of data privacy.

E1.2.4 DATA TRANSFER BETWEEN COMMUNICATING ENTITIES

The transfer of data formatted in compliance with this Recommended Standard between communicating entities should be accomplished via secure mechanisms approved by the IT Security functionaries of exchange participants.

E1.2.5 CONTROL OF ACCESS TO RESOURCES

This Recommended Standard is not directly involved in the control of access to resources. This Recommended Standard assumes that control of access to resources will be managed by the systems upon which packaging and parsing of packages compliant to the Recommended Standard are performed.

E1.2.6 AUDITING OF RESOURCE USAGE

This Recommended Standard is not directly involved in the management or auditing of resource usage. This Recommended Standard assumes that the management of systems upon which this Recommended Standard is implemented will handle auditing of resource usage.

E1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

There are no unique threats or attack scenarios that apply specifically to the technologies specified in this Recommended Standard. The management of those systems and networks should address potential threats or attack scenarios applicable to the systems and networks on which this Recommended Standard is implemented.

E1.4 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The consequences of not applying security to the systems and networks on which this Recommended Standard is implemented could include potential loss, corruption, and theft of data and loss of data privacy.

E1.5 DATA SECURITY IMPLEMENTATION SPECIFICS

XML was chosen as the concrete implementation language for Descriptors and XML-based XFDU was chosen as the SIP implementation so any XML-related security technologies could be used by a system. XML-related security technologies are being developed in a number of other organizations. Users and implementers of this standard should adhere to those standards, as appropriate.

The XFDU SIP Implementation allows the use of the XFDU standard data structures for the inclusion of data object checksums and the description of any encryptions that have been to be applied to Data Objects.

E2 SANA CONSIDERATIONS

E2.1 SANA REGISTRY

The recommendations of this document request SANA to create a registry named “Data Archive Ingest XML” that consists of a set of XML schemas. The initial registry should be populated with the schema and XML detailed in annex A of this document.

The registration rule for an addition or a change to this registry requires publication of a new CCSDS document or a new issue of a CCSDS document.

E2.2 CONTROL AUTHORITY OFFICE REGISTRY

A Control Authority Office registry (already existing within the SANA registry structure) contains an entry for the ‘CCSD’ Control Authority Office. Each Control Authority Office in turn handles registration of formats and assigns Authority and Description Identifiers (ADIDs) for registrations under its authority.

This Recommended Standard will result in requests to the ‘CCSD’ Control Authority Offices to update their ADID registry in accordance with their procedures to add the following four new records.

ADID	Description
CCSD0014	Transfer Object Type Descriptor
CCSD0015	Collection Descriptor
CCSD0016	Constraints on SIPs
CCSD0017	Abstract SIP

E3 PATENT CONSIDERATIONS

There are no known patent issues related to this Recommended Standard.

ANNEX F

SIP TO XFDU MAPPING EXAMPLE

(INFORMATIVE)

F1 EXAMPLE DESCRIPTION

The following example is a very simple concrete implementation of a SIP. The objective is to show how in practice the SIP elements map into the XFDU extension elements.

The following example conforms to the XFDU standard that is implemented in this case as a zip file.

The objective of this SIP is the delivery of one PDF file from a scientific laboratory, the LESIA, to the CNES CDPD Archive (Plasma Physics Data Center). This PDF file is the description of the WAVES experiment of the WIND mission. This file is included in the SIP.

This PDF file ‘waves_documentation.pdf’ is a Data Object described by the Transfer Object Type Descriptor ‘WavesDocTransferObject.xml’ given in F3.

F2 SIP STRUCTURE

The following SIP package is a zip file that contains:

- a) the pdf file;
- b) the XFDU manifest.

The following XFDU manifest is made up of:

- The package header, which includes the SIP Global Information.
- The information package map, which incorporates three nested content units. Each Content Unit maps correspondingly to the following three SIP concepts: SIP Transfer Object, SIP Transfer Object Group, and SIP Data Object. Within the Content Unit for the SIP Data Object, the internal link ‘dataObject1’ is a pointer into the manifest’s Data Object section. This section specifies the location of the delivered file which can be seen to be within the zip file.
- The data object section, which includes the link to the data file itself ‘waves_documentation.pdf’.

The associated XFDU manifest ‘WavesDocManifest.xml’ is given in F4.

The links between the SIP and the Transfer Object Type Descriptor are made by the IDs 'producerSourceID =LESIA, descriptorID =waves_documentation, transferObjectID =cdpp-wind-transfer-object-0020, associatedDescriptorGroupTypeID =G1, associatedDescriptorDataID =TNR_L2_DOC'.

F3 WAVES TRANSFER OBJECT TYPE DESCRIPTOR

```

<?xml version="1.0" encoding="UTF-8"?>
<transferObjectTypeDescriptor xmlns="urn:ccsds:schema:pais:1">
  <identification>
    <descriptorModelID>CCSD0014</descriptorModelID>
    <descriptorModelVersion>V1.0</descriptorModelVersion>
    <descriptorID>WAVES_DOCUMENTATION</descriptorID>
    <producerSourceID>LESIA</producerSourceID>
  </identification>
  <description>
    <transferObjectTypeTitle>TNR documentation</transferObjectTypeTitle>
    <transferObjectTypeDescription>Textual documents describing the Waves
experiment</transferObjectTypeDescription>
    <transferObjectTypeOccurrence>
      <minOccurrence>1</minOccurrence>
      <maxOccurrence>1</maxOccurrence>
    </transferObjectTypeOccurrence>
  </description>
  <relation>
    <parentCollection>WAVES_DESCRIPTION_CO</parentCollection>
    <association>
      <targetID>WIND_WAVES_CO</targetID>
      <relationDescription>
        <relationType>description</relationType>
      </relationDescription>
    </association>
  </relation>
  <groupType>
    <groupTypeID>G1</groupTypeID>
    <groupTypeStructureName>set</groupTypeStructureName>
    <dataObjectType>
      <dataObjectTypeID>TNR_L2_DOC</dataObjectTypeID>
      <dataObjectTypeDescription>Textual document describing the Waves experiment: measurement
principle and instrument description.</dataObjectTypeDescription>
      <dataObjectTypeOccurrence>
        <minOccurrence>1</minOccurrence>
        <maxOccurrence>1</maxOccurrence>
      </dataObjectTypeOccurrence>
      <dataObjectTypeFormat>
        <mimeType>application/pdf</mimeType>
        <registrationInformation>
          <registrationAuthority>authReg</registrationAuthority>
          <registeredID>IdReg</registeredID>
        </registrationInformation>
      </dataObjectTypeFormat>
    </dataObjectType>
  </groupType>
</transferObjectTypeDescriptor>

```

F4 WAVES XFDU MANIFEST IMPLEMENTATION

```

<?xml version="1.0" encoding="UTF-8"?>
<xfdu:XFDU xmlns:xfdu="urn:ccsds:schema:xfdu:1" xmlns:pais="urn:ccsds:schema:pais:1">
  <packageHeader ID=" cdpp-wind-sip-0020">
    <volumeInfo>
      <specificationVersion>1.0</specificationVersion>
    </volumeInfo>
    <environmentInfo>
      <extension>
        <!-- SIP Global attributes -->
        <pais:sipGlobalInformation>
          <pais:sipID>cdpp-wind-sip-0020</pais:sipID>
          <pais:producerSourceID>LESIA</pais:producerSourceID>
          <pais:producerArchiveProjectID>cdpp-wind</pais:producerArchiveProjectID>
          <pais:sipContentTypeID>SIP-TYPE-01-EXPERIMENT-
DESCRIPTION</pais:sipContentTypeID>
          <pais:sipSequenceNumber>0020</pais:sipSequenceNumber>
        </pais:sipGlobalInformation>
      </extension>
    </environmentInfo>
  </packageHeader>

  <informationPackageMap>
    <!--From here follow nested Content Units for Transfer Object content description-->

    <xfdu:contentUnit textInfo="Waves documentation">
      <!--Transfer Object Type description-->
      <extension>
        <pais:sipTransferObject>
          <pais:descriptorID>WAVES_DOCUMENTATION</pais:descriptorID>
          <pais:transferObjectID>cdpp-wind-transfer-object-0020</pais:transferObjectID>
        </pais:sipTransferObject>
      </extension>

      <xfdu:contentUnit>
        <!--Transfer Object Group (inside Transfer Object) description-->
        <extension>
          <pais:sipTransferObjectGroup>
            <pais:associatedDescriptorGroupTypeID>G1</pais:associatedDescriptorGroupTypeID>
            <pais:transferObjectGroupInstanceName>Group1</pais:transferObjectGroupInstanceName>
          </pais:sipTransferObjectGroup>
        </extension>

        <xfdu:contentUnit>
          <!--Data Object (inside Transfer Object Group) description-->
          <extension>
            <pais:sipDataObject>
              <pais:associatedDescriptorDataID>TNR_L2_DOC</pais:associatedDescriptorDataID>
              <pais:dataObjectPreservationName>preservedName</pais:dataObjectPreservationName>
            </pais:sipDataObject>
          </extension>
          <!--Byte Stream (inside Data Object) description-->
          <dataObjectPointer dataObjectID="dataObject1"/>
        </xfdu:contentUnit>
      </xfdu:contentUnit>
    </xfdu:contentUnit>
  </informationPackageMap>
</xfdu:XFDU>

```

```
</xdu:contentUnit>
</xdu:contentUnit>
</informationPackageMap>

<dataObjectSection>
  <dataObject ID="dataObject1">
    <byteStream mimeType="application/pdf">
      <fileLocation locatorType="URL" href="file:datafiles/waves_documentation.pdf"/>
      <checksum checksumName="MD5">0d81f6a7ba42eef4b1bf1e694ca59d1</checksum>
    </byteStream>
  </dataObject>
</dataObjectSection>

</xdu:XFDU>
```