The Consultative Committee for Space Data Systems

Report Concerning Space Data System Standards

OPEN ARCHIVAL INFORMATION SYSTEM INTEROPERABILITY FRAMEWORK (OAIS-IF) RATIONALE, SCENARIOS, AND REQUIREMENTS

DRAFT INFORMATIONAL REPORT

CCSDS 000.0-G-0

DRAFT GREEN BOOK

January 2021

AUTHORITY

Issue:	Draft Green Book, Issue 0
Date:	January 2021
Location:	Not Applicable

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

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical experts from 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-4).

This document is published and maintained by:

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

FOREWORD

[Foreword text specific to this document goes here. The text below is boilerplate.]

Commentato [MWK1]: Should be the three paragraphs that are in the ADD Foreword, I think.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This document 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-4). 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 (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- 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 and Issue	Date	Status
CCSDS 000.0-G-0	OPEN ARCHIVAL INFORMATION SYSTEM INTEROPERABILITY FRAMEWORK (OAIS-IF) RATIONALE, SCENARIOS, AND REQUIREMENTS, Draft Informational Report, Issue 0	January 2021	Current draft

CONTENTS

Contents

Table of Figures

1 INTRODUCTION

THIS DOCUMENT CONTAINS LINKS TO OTHER DOCUMENTS ON THE WEB, IN ORDER TO KEEP THE INFORMATION AVAILABLE.

EVENTUALLY THESE LINKS WILL BE DELETED.

1.1 PURPOSE

The purpose of this document is to describe the rationale, with some motivating scenarios, and requirements for the CCSDS and International Organization for Standardization (ISO) Open Archival Information System Interoperability Framework (OAIS-IF). It is a supplement to the overarching OAIS [1] Reference Model standard.

The purpose of the set of standards which make up OAIS-IF is to define interfaces and services which go beyond the exchange of data (as in communications standards) and instead will enable the exchange of information.

The difference may be understood as follows. Following OAIS, data, and in particular digital data, which OAIS calls a Digital Object, is simply an object composed of a set of bit sequences. Bit sequences by themselves could mean anything. On the other hand, Information is *any type of knowledge that can be exchanged. In an exchange, it is represented by data*. It is made up of the Data Object plus what OAIS refers to as Representation Information.

The aim is to improve interoperability between users and archives by transferring Information rather than just Data, and to enable the increased usability of information of all types across all domains.

1.2 SCOPE

The OAIS-IF adds capabilities for system interoperability between more general Consumers and Producers and archives, whether the archive is OAIS Reference Model conformant or not.

OAIS-IF does not define what should happen within an archive or user systems except as needed to define an interoperable interface. The scope is to describe how information is transferred between users and archives in such a way that the information is understandable, as far as possible, and also, if required, can be accompanied by evidence about its Authenticity.

1.3 APPLICABILITY

OAIS-IF should be applicable to any archive and any user. For an OAIS compliant archive, OAIS-IF should enable the archive to make not only complete AIPs but also all of the individual components of its AIPs separately available. A non-OAIS compliant archive may

Commentato [MWK2]: Interfaces may be other than APIs.

Commentato [DG3R2]: The original text was “APIs and services” to cover these extra things

Commentato [MWK4]: I thought OAIS-IF was going beyond the purposes of the OAIS RM (Trustworthiness) and adding interoperable access to information. Doesn’t the OAIS RM allow bespoke exchange of information with RepInfo describing the data?

Commentato [DG5R4]: This is a comparison with other standards about communications.
OAIS does not say much specifically about exchanges.

Commentato [MWK6R4]: Ah, I see. I’ll make a little addition to hopefully help clarify.

Commentato [MWK7]: To reiterate… doesn’t the OAIS RM allow the transfer of information also (within the DC), just not with interoperable standards? And also, in some cases, for users outside the Designated Community?

Commentato [DG8R7]: OAIS “allows” it but does not say how to do it.

Commentato [MWK9R7]: I think this is answered by the above exchange.

Commentato [MWK10]: Seems like when we use the word “conformant” we should specifically say “Reference Model.”

Commentato [DG11R10]: OK

Commentato [MWK12]: Suggest “may be applied” rather than should be applicable. As written, it sounds like most existing archives are already OAIS-IF compliant.

Commentato [DG13R12]: We are talking about applicability i.e. were it could be used, not where it IS being used.
The point is that we are designing OAIS-IF to be broadly usable.

Commentato [MWK14R12]: I agree with your intent, but I think “should be” carries a different implication. To adopt your word (could) into my suggestion, how about “**could be applied to any archive**…” That eliminates the implication that we could be talking about existing implementations. As in “blue uniforms should be applicable to any police department” (including existing ones.)

Commentato [MWK15]: It’s already making AIPs (or their content as manifest in DIPs) available to the Designated Community that subscribes to its bespoke interface. The difference with OAIS-IF is broader accessibility for more users.

Commentato [DG16R15]: The point is that OAIS-IF APIs will allow all the components to be obtained individually, rather than as full AIPs.
The DIP definition does not specify anything much.

Commentato [MWK17R15]: With that clarification, I’ve removed my earlier redline, and added one that (I think) says what you meant. Correct?

However, I still think this is an incongruous usage of “AIPs” since customers only see DIPs and SIPs, not AIPs. But I can live with it.

not have all the components required for an AIP available but OAIS-IF can still be used in a more limited way.

1.4 RATIONALE

The rationale and vision for OAIS-IF is that in the long-range future it will provide:

- A common user interface experience for users (providers and consumers) of OAIS Archives when accessing many diverse kinds of archives through the OAIS-IF.
- An efficient standardized way for archives to exchange data between archives using the same standardized OAIS-IF interfaces.
- Given broad acceptance of OAIS-IF in the OAIS community, a better chance that long-term preservation will work because future generations can easily find the interfacing resources (plug-ins, etc.) that can be used to access legacy archives.
- Enhanced capabilities for cross-discipline research when many different disciplines use the same interface, and access to a new archive outside of their Designated Community can be accomplished via OAIS-IF.

Digitally encoded information is very important. The OAIS standard tells us how to preserve that information so that it will continue to be usable in the long term by a Designated Community. However, each Archive will have different Designated Communities. Many users of such information will not be members of any Designated Community. In that case, while their ability to understand the information is not guaranteed, OAIS and OAIS-IF still support their ability to access it, with as much Representation Information as is available. Moreover, not all archives are OAIS conformant (as defined in the OAIS Reference Model section 1.4), yet they too contain important information. While archives that are not OAIS Reference Model conformant were not the drivers of OAIS-IF development, OAIS-IF should still be able to provide significant interoperability for most of them.

Those users within one domain, such as Astronomy, tend to have common software tools and common terminology, algorithms etc., and so are likely to be able to understand and use data from various sources, and in particular from other Astronomical archives. A person outside that community will have much greater difficulty in using such data.

OAIS-IF seeks to make the information from different domains more easily usable for a broader range of customers. It will be of general benefit if all can exchange, understand and use information from all domains; it will also be of benefit to archives by increasing the use of their holdings. The OAIS-IF set of documents are designed for use by archives, information creators and users.

OAIS-IF does not seek to specify how an archive preserves information, nor how information should be created and encoded, nor what software users must employ when using information. Instead OAIS-IF defines how information should be exchanged if that information is to be usable. In addition, OAIS-IF allows Preservation Description

Information (PDI) to be exchanged, and in particular Provenance Information, with which a user can judge Authenticity.

1.5 CONTEXT

Commentato [MWK18]: In the publications manual, this section (as written) seems to be titled DOCUMENT STRUCTURE. Probably we should follow that convention.

Section 2 provides a brief overview of OAIS-IF and its uses. A number of scenarios are described in section 3; in each scenario issues, avoiding repetition, are identified. Based on these scenarios, and other considerations, a number of requirements are listed in section 4, with cross-references to the scenarios, where appropriate. Further details of the set of books which define the OAIS-IF are provided in section 6.

1.6 DEFINITIONS

1.6.1 ACRONYMS AND ABBREVIATIONS

AIP	Archival Information Package
CCSDS	Consultative Committee for Space Data Systems
DEDSL	Data Entity Specification Language
DIP	Dissemination Information Package
FITS	Flexible Image Transport System
GIS	Geographic Information System
ISO	International Organization for Standardization
OAIS	Open Archival Information System
PDI	Preservation Description Information
SIP	Submission Information Package
TEI	Text Encoding Initiative
UML	Unified Modeling Language
XML	Extensible Markup Language

1.6.2 TERMINOLOGY

Digital preservation interests a range of different communities, each with a distinct vocabulary and local definitions for key terms. A glossary is included in this document, but it is important to draw attention to the usage of several key terms.

In general, key terms in this document have been adopted from the OAIS Reference Model. One of the great strengths of the OAIS Reference Model has been to provide a common terminology made up of terms ‘not already overloaded with meaning so as to reduce conveying unintended meanings’ (reference [1]). Because the OAIS has become a foundational document for digital preservation, the common terms are well understood and are therefore used within this document.

The OAIS Reference Model uses ‘digital archive’ to mean the organization responsible for digital preservation. In this document, the term ‘repository’ or phrase ‘digital repository’ is used to convey the same concept in all instances except when quoting from the OAIS. It is

important to understand that in all instances in this document, ‘repository’ and ‘digital repository’ are used to convey digital repositories and archives that have, or contribute to, long-term preservation responsibilities and functionality. This document uses the OAIS concept of the ‘Designated Community’. A repository may have a single, generalized ‘Designated Community’ (e.g., every citizen of a country), while other repositories may have several, distinct user communities with highly specialized needs, each requiring different functionality support from the repository; this document uses the term Designated Community to cover this second case also.

1.6.2.1 Glossary

Unless otherwise indicated, other definitions are taken from the OAIS Reference Model (reference [1]) and are provided here for convenience.

Access Policy: Documented statement, authorized by the repository management, that describes the approach to be taken by the repository for providing access to objects accessioned into the repository. The Access Policy may distinguish between different types of access rights, for example between system administrators, members of the Designated Community, and general users.

Archival Information Package (AIP): An Information Package, consisting of the Content Information and the associated Preservation Description Information (PDI), which is preserved within an OAIS.

Authenticity: The degree to which a person (or system) regards an object as what it is purported to be. Authenticity is judged on the basis of evidence.

Consumer: The role played by those persons, or client systems, who interact with OAIS services to find preserved information of interest and to access that information in whatever level of detail is allowed. In addition to the normally expected entities outside the OAIS, this can also include other OAISes, as well as internal OAIS persons or systems.

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

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

Fixity Information: The information which documents the mechanisms that ensure that the Content Data Object has not been altered in an undocumented manner.

Information: Any type of knowledge that can be exchanged. In an exchange, it is represented by data.

NOTE – An example of Information is a string of bits (the data) accompanied by a description of how to interpret the string of bits as numbers representing temperature observations measured in degrees Celsius (the Representation Information).

Information Object: A Data Object together with its Representation Information.

Information Package: A logical container composed of optional Information Object(s). Associated with this Information Package is Packaging Information used to delimit and identify the Information Object and optional Package Description information used to facilitate searches for the Information Object.

Other Representation Information: A type of Representation Information which cannot easily be classified as Structure Representation Information or Semantic Representation Information. It is a type of Information Object.

NOTE – For example, software, algorithms, encryption, written instructions and many other things may be needed to understand the Content Data Object in ways exemplified by the Preservation Objectives, all of which therefore would be, by definition, Representation Information, yet would not obviously be either Structure Representation Information or Semantic Representation. Information defining how the Structure Representation Information and the Semantic Representation Information relate to each other, or software needed to process a database file would also be regarded as Other Representation Information.

Packaging Information: The information that describes how the components of an Information Package are logically or physically bound together and how to identify and extract the components. It is a type of Information Object.

Physical Object: An object (such as a moon rock, bio-specimen, microscope slide) with physically observable properties that represent information that is considered suitable for being adequately documented for preservation, distribution, and independent usage.

Preservation Description Information (PDI): The information, which along with Representation Information, is necessary for adequate preservation of the Content Data Object and which can be categorized as Provenance Information, Context Information, Reference Information, Fixity Information, and Access Rights Information. It is a type of Information Object.

NOTE – Defining PDI (as well as its components: Provenance Information, Context Information, Reference Information, Fixity Information, and Access Rights Information) as relevant to the Content Data Object does not mean that those concerns are any less important for other data objects or at other levels, for example, it is important to apply reference, fixity, provenance, context and access rights to Representation Information, or to any other information the Archive is preserving. Definition of these terms as relevant to the Content Data Object is simply to ease discussion of these concepts at the Content Data Object level.

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

Provenance Information: The information that documents the history of the Content Data Object. This information tells the origin or source of the Content Data Object, any changes that may have taken place since it was originated, and who has had custody of it since it was originated. The Archive is responsible for creating and preserving Provenance Information from the point of Ingest; however, earlier Provenance Information should be provided by the Producer. Provenance Information adds to the evidence to support Authenticity.

Reference Information: The information that is used as an identifier for the Content Data Object. It also includes identifiers that allow outside systems to refer unambiguously to a particular Content Data Object.

NOTE – An example of Reference Information is an ISBN.

Representation Information: The information that maps a Data Object into more meaningful concepts so that the Data Object may be understood in ways exemplified by Preservation Objectives. It is a type of Information Object.

NOTE – An example of Representation Information for a bit sequence which is a FITS file might consist of the FITS standard which defines the format plus a dictionary which defines the meaning in the file of keywords which are not part of the standard. This would then allow the information in the FITS file to be used by a computer program to display the image which may be contained in the FITS file, together with the associated coordinate system so that a human can identify objects of interest, for example stars or galaxies. Alternatively, the computer program may identify such objects automatically.

Representation Information Network: The set of Representation Information that fully describes the meaning of a Data Object. Representation Information in digital forms needs additional Representation Information so its digital forms can be understood over the Long Term. It is a type of Information Object.

Semantic Representation Information: The Representation Information that further describes the meaning of the Data Object, and its parts or elements, beyond that provided by the Structure Representation Information.

NOTE – For example, Semantic Representation Information may describe the meaning of columns, and perhaps particular values seen in the columns of a spreadsheet.

Structure Representation Information: The Representation Information that imparts information about the arrangement of and the organization of the parts or elements of the Data Object.

NOTE – For example, Structure Representation Information maps bit streams to common computer types such as characters, numbers, and pixels and aggregations of those types such as character strings and arrays.

Submission Agreement: The agreement reached between an OAIS and the Producer that specifies the intended formats and content descriptions, and any other arrangements needed

(such as delivery or transmission protocols), for the Data Submission Sessions. Such specifications establish the intended deliverables from the Producer and how they are represented on each delivery through physical media or in a telecommunication dialogue.

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.

1.7 REFERENCES

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

[A list of documents referenced in the report goes here. See CCSDS A20.0-Y-4, *CCSDS Publications Manual* (Yellow Book, Issue 4, April 2014) for reference list format.]

[1] *Reference Model for an Open Archival Information System (OAIS)*. Recommendation for Space Data System Practices, CCSDS 650.0-M-2. Blue Book. Issue 1. Washington, D.C.: CCSDS, June 2012. [Equivalent to ISO 14721:2012.], or later Available from: https://public.ccsds.org/Pubs/650x0m2.pdf . The definitions are taken from the draft update of OAIS.

For the PPT of the diagrams see https://www.dropbox.com/s/c3ribtzpyz6ynh9/OAIS-IF-diagrams.pptx?dl=1

2 OVERVIEW

We are interested in transferring usable information between two or more entities.

Commentato [MWK19]: This sentence seems out-of-context. Maybe just needs to be staged better. Something like "In the following scenario explored in this section…" Or "While section 3 explores specific user scenarios, this section (2) will look at the broadest overview of OAIS-IF scenarios."

2.1 CURRENT SITUATION

A current example of transferring information between entities is a client-server system, for example a web browser and a web server, where what is being transferred between the two is described by HTML, with perhaps MIME encoded objects such as images, using the HTTP protocol via TCP/IP. However, this is specific to HTML and the Representation Information required to deal with the HTML and a limited number of MIME types is embedded in the web browser software.

A web browser would not be able to deal with scientific data generally except to allow it to be downloaded as a collection of bits, and as such would not be usable without a great deal of effort on the part of the user, for example to identify then read documentation and download and install software. The user may not even know where to get the documentation and software.

2.2 WHAT OAIS-IF SHOULD MAKE POSSIBLE

OAIS-IF defines a set of common interfaces and functionality which will allow developers to create interfacing software which in turn provides users with access to information in an archive or set of archives. The software developers may or may not be associated with the archive, but could, for example, be supporting efforts in user groups. Using that software, consumers will be able access data and associated Representation Information (if available) from archives across disparate domains of study.

Some of the interfaces will be generally applicable, and reference implementations should be available. The implementation of other interfaces will be specific to the software being used by the archive or information creator or information user.

The following illustrates the way in which OAIS-IF will enable increased usage between two generic pieces of software, shown here as S1 and S2. The "OAIS-IF interfaces and objects" is based on the OAIS Information Model, which itself is designed to be applicable to any type of Information. The adapters allow S1 and S2 to use the OAIS-IF interfaces and objects.

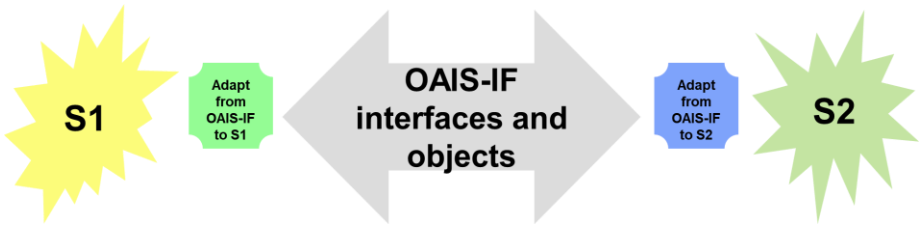


Figure 2-1 General use of OAIS-IF

The advantage of using OAIS-related ideas is that the OAIS Information Model was designed to accommodate any type of digitally encoded Information in a way which allows that information to be usable.

This can then be generalized to show the way in which users can use their software to access and use information from multiple different archives for which their software is not originally designed.

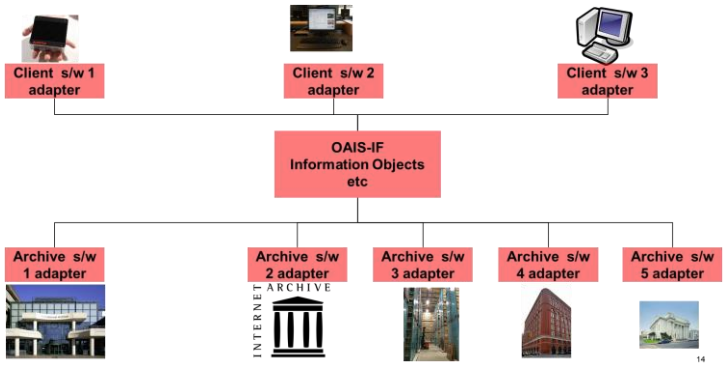


Figure 2-2 Connections between clients and archives using OAIS-IF

This has the advantage that each client only needs one adapter, specific to the client software being used, which allows that client to accept and use the OAIS-IF Information Objects, independent of the archive with which it is communicating, as long as each of those archives, with their adapters, use OAIS-IF Information Objects.

Therefore, the client with its single adapter can get, and understand, information from multiple archives simultaneously.

An alternative way one might think to implement this would be to find a common format into which all other data could be transformed. However;

1. Format by itself is not adequate. OAIS uses the term Structure Representation Information as a general way of talking about "format", but it recognises that one also needs semantics (Semantic Representation Information") and "Other Representation Information" such as software.
2. Those formats which are extremely flexible such as HDF5[1] or HDS[2] require that the software which uses it is specifically tailored to the way in which the structure of these files is arranged.
3. A common format has often been sought, and indeed many formats have the word "Common" or even "Universal" in their name. Moreover, experience shows that such

Commentato [MWK20]: In the spirit of what I *think* we've been discussing, it may be that a client will need an adapter specific to the archive features. This picture seems to preclude that. Maybe a footnote that says "Archive SW adapters may be at either the client or archive locations." (?)

Commentato [DG21R20]: It is important to read the full text in this section.

Commentato [MWK22R20]: I did the first time. If I'm missing a point, please illuminate me. Seems like my footnote should still apply, if we are sticking by that "adapter can be anywhere" approach.

On second thought, never mind. Let's not say it and we won't have to revise it if we go back to the original approach.

[1] https://www.hdfgroup.org/solutions/hdf5/

[2] See http://starlink.eao.hawaii.edu/docs/sun92.htx/sun92.html

transformations always lose information, and formats named "Common" or "Universal" fail to live up to their names.

4. The sheer size of the data may preclude such transformations in any practical way.

Therefore OAIS-IF must be designed to follow the more general OAIS Information Model, so that OAIS Information Objects are transferred, hence explaining why this is called **OAIS**-IF. The advantage of using Information Objects is that they can contain any digitally encoded information but are required to have associated Representation Information, which allows the information encoded in the digital object to be understood – this is discussed in more detail below.

In order to provide a little more detail Figure 2-1can be expanded as follows.

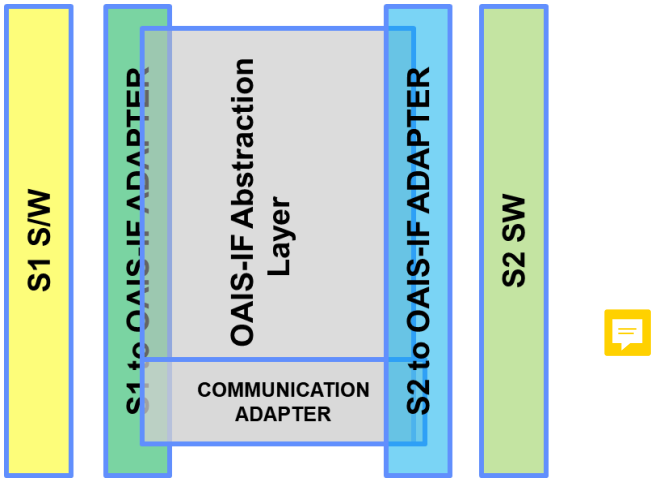


Figure 2-3 Expanded view of use of OAIS-IF

The S1 s/w is, for example, existing software which probably was created without knowledge of OAIS-IF.

In order integrate with OAIS-IF one needs an adapter, called here the "S1 to OAIS-IF adapter" or "S1 adapter" for convenience.

The "S1adapter" must be software tailored to the S1 software and must also be software which implements the OAIS-IF interfaces (shown here as the OAIS-IF Abstraction Layer) and required functionality.

The adapter by itself may not be able to use the data and Representation Information it initially receives, and therefore it requires additional Representation Information either from the same source or from different sources. It is also possible that there simply is not enough Representation Information available to make the information received usable by the S1

software. For example, the S1 software may only be able to display images and so will never be able to deal with scientific tables, unless the Representation Information essentially does something additional, such as displaying an additional window, independent of the S1 software's image display, which can deal with such tables.

At the very least, the Representation Information may simply be pointers to documentation for the user to read, or pointers to software which is tailored to the data and which needs to be installed separately.

The adapter software includes parts which implement the communications between S1 and S2, either embedded within OAIS-IF or as a separate set of interfaces used by the S1 adapter (shown here as the Communications Adapter). Communications software already exists, and various frameworks provide convenient ways to support multiple communications stacks without changing the adapters.

In a symmetrical way the "S2 to OAIS-IF adapter" (or S2 adapter for short) must be software tailored to the S1 software and must also be software which implements the OAIS-IF interfaces and required functionality.

Of course, if S1 (or S2) already implements OAIS-IF then the associated adapter is not needed, but this special case does not cause any problems.

Note that arrangement of this diagram in layers is to show that each layer only communicates to its adjacent layers, through well-defined interfaces.

The layers do not specify **where** those layers are placed. Other diagrams will show potential placements of the software – see Figure 7-4 and Figure 7-5.

2.3 EXAMPLE IMPLEMENTATION OF AUTOMATICALLY USABLE REPRESENTATION INFORMATION FOR MULTIDISCIPLINARY WORK

To provide a more concrete example which also show the use of the single adapters and the Representation Information, Figure 2-4 illustrates a user's software S1 which gets information from two archives which use software S2 and S3.

One type of Representation Implementation which can be used quite easily is as serialized Java objects in a Java implementation. Such as object can be de-serialised to create a Java Object which can be examined through the Java Reflection API to discover which interfaces this object implements. This types of Representation Information is used in the example below.

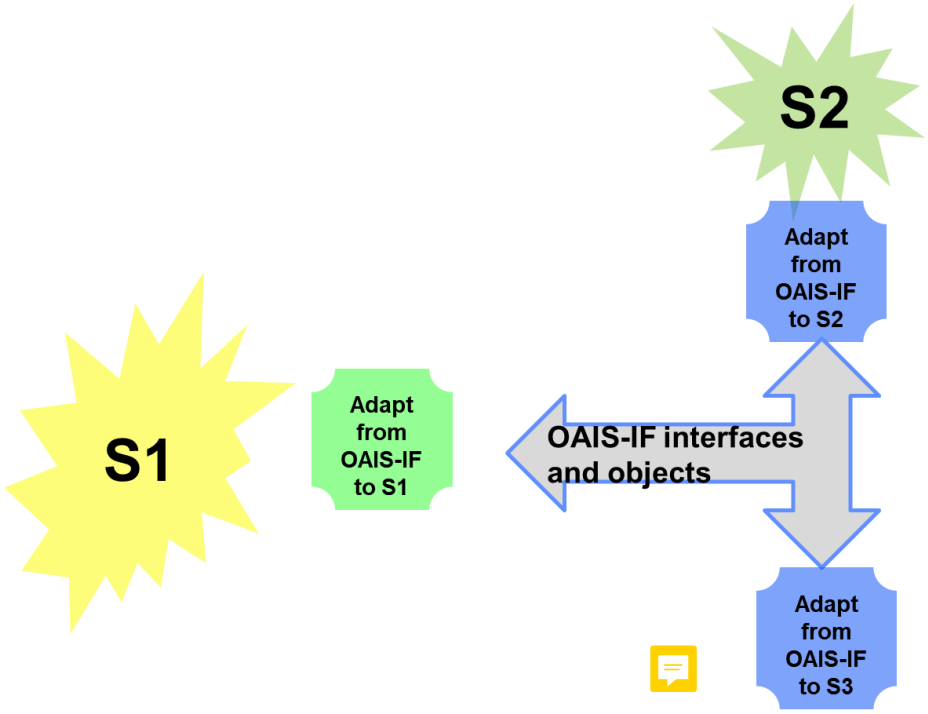


Figure 2-4 Multi-archive example

For example, if one is using client software (S1) which can use JTable interfaces and which, via the adapter, receives information from an archive containing FITS files (S2). The information is in the form of an Information Object as defined by OAIS-IF.

Information may also be obtained from S3, with its own adapter. In this case the archive contains tables in the form of CSV files.

The adapter associated with S2 constructs and sends the Information Object. The adapter associated with S1 can separate the Data Object, which in this example is a FITS file which contains a FITS table, and Representation Information. The Representation Information in this example is a serialized Java object which when de-serialised in the adapter becomes a Java object which the adapter can discover implements an interface which can read FITS tables as well as the Java JTable interface. The adapter is tailored to the client software and therefore is built to be able to provide the JTable objects to S1.

In a similar way the adapter for S3 constructs Information Objects consisting of a Data Object (the CSV file) and the associated Representation Information (the Structure RepInfo which described the UTF-8 and CSV encoding, and the Semantic RepInfo which describes

the names, meanings and units of the columns plus Other RepInfo which in this case is again a serialised Java Object). As with the Information Objects from S2, the S1 adapter separates the Data Object from the Representation Information. De-serialising the Other RepInfo reveals that the Java Object also implements the Java JTable interface, using the Data Object, Structure RepInfo and Semantic RepInfo.

In exactly the same way as with the Information Objects from S2, the S1 adapter provides JTable Objects to S1.

Having JTables both from archives S2 and S3 the Information User of S1 can then combine, compare the information from these two sources.

In this way each of S1, S2 and S3 have their own unique adapters while the Representation Information allows the information to be combined for multidisciplinary work.

It can be seen that the same will apply whether S1 obtains Information Objects from any number of archives.

3 OAIS-IF SCENARIOS

The following provides a limited set of user scenarios which OAIS-IF will support.

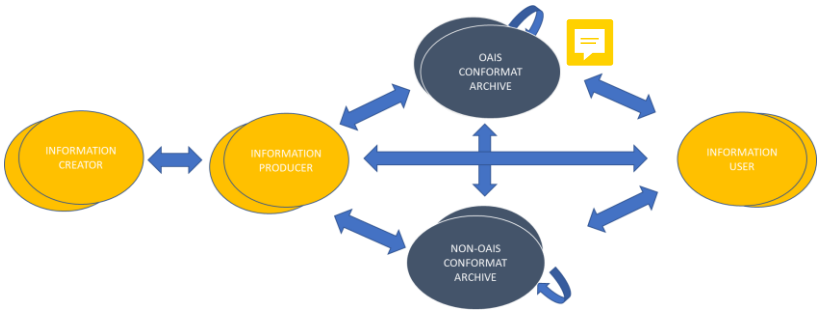


Figure 3-1 Roles of entities exchanging information

Information of various types is exchanged between individuals or organisations or systems which may be fulfill one or more of the following role

- Information Creators
- Information Producers
- Information Users, some of whom may be members of one or more Designated Communities of one or more OAIS conformant archives.

  NOTE: OAIS uses the term "Consumers" to refer to the role played by those persons, or client systems, who interact with OAIS services. Therefore, Consumers are a subset of Information Users because the latter may interact with services which are not connected with an OAIS.
- OAIS conformant archives

  NOTE: Archive-to-archive interfaces (for both OAIS-conformation archives and non-OAIS-conformant archives) may be specified in this document (and accomplished by implementers) by using "Producer" and "Consumer" interfaces rather than by specific interfaces for transactions between archives.
- Archives which are not OAIS conformant

In the OAIS-IF set of documents all these exchanges of information are assumed to be encoded in one or more Data Objects, with associated Representation Information. It is reasonable to propose that these are packaged together into OAIS Information Packages (Figure 3-2).

Commentato [MWK23]: A publication manual question, maybe… Shouldn't all material under a section (like 3) be under a subsection (like 3.1)? Otherwise readers would have to refer to "that stuff that is in section 3 but before section 3.1…"

Commentato [DG24R23]: Yes it is true that Tom has been inserting subsection numbers at the start.

Commentato [MWK25]: Producers also interact with OAIS services, don't they? I expected "Consumers" to be those persons/clients who receive archived information and/or it's associated representation information (in DIPs.)

Commentato [DG26R25]: I am just explaining why Information Users is being used – because the definition of Consumers in OAIS is more restrictive.

Commentato [MWK27R25]: OK, got it.

Commentato [MWK28]: With my additional parenthetical, I recommend moving this below the bottom bullet so it would explain both above it.

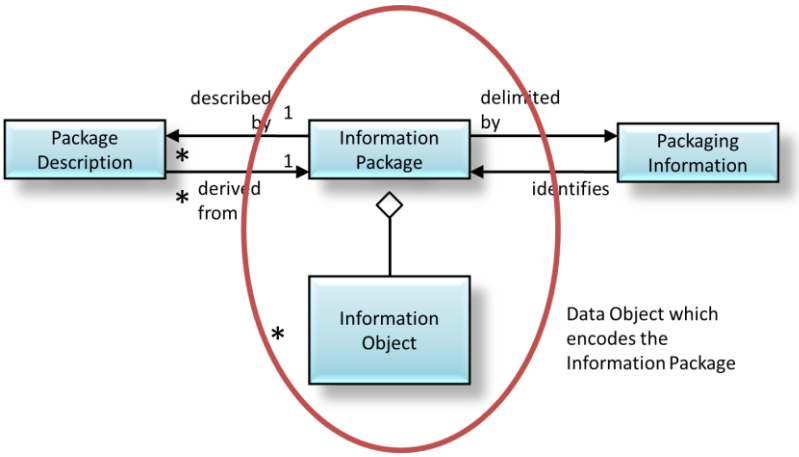


Figure 3-2 OAIS Information Package

The Information Packages are themselves made up of sets of bits, in other words Data Objects. These must be associated with their own Representation Information, of which, for the purposes of this document, Packaging Information is a part. The Information Packages will be transmitted between the entities by communication protocols such as TCP/IP, CCSDS DTN etc.

The full Archival Information Package (AIP) identifies all the pieces of information required to preserve the Content Information is shown in Figure 3-3. OAIS-IF should support all these components of the AIP. However, this does not imply that all information sources, nor all transfers, use AIPs. The only requirement is that the design of OAIS-IF should support all the component objects required by an AIP.

For an OAIS Archive, the transferred Information Packages should have adequate content to allow an OAIS archive to build AIPs. For non-OAIS archives, other transactions (submission agreements, etc.) between the users and the archive will have to establish the needed content of the transferred Information Packages.

Commentato [MWK29]: Shouldn't this be "contains" rather than identifies? Identifying presumes the existence of a reader/client, and instead the AIP is data at rest.

Commentato [DG30R29]: OAIS says the AIP logically contains, which I have rephrased as "identifies".

Commentato [MWK31R29]: I agree with OAIS, and still think it should be contains rather than identifies. Different functions, not synonomous.

Commentato [MWK32]: Transfers use DIPs and SIPs. This is a negative statement ("they don't use AIPs"). Seems like it should at least be accompanied by a positive statement (what do they use?).

Commentato [DG33R32]: DIP is just a name. OAIS does not say that a DIP cannot be the same as an AIP. The point here is to make it clear that OAIS-IF allows one to get just pieces of the contents of the archive.

Commentato [MWK34R32]: OK.

Commentato [MWK35]: Incoming DIPs, right? Why not say that?

Commentato [DG36R35]: OAIS defines **Dissemination Information Package (DIP)**: An Information Package, derived from one or more AIPs, and sent by Archives to the Consumer in response to a request to the OAIS.

The point was made above that we use the term "Information User" is broader than the OAIS definition of "Consumer"

Commentato [MWK37R35]: OK.

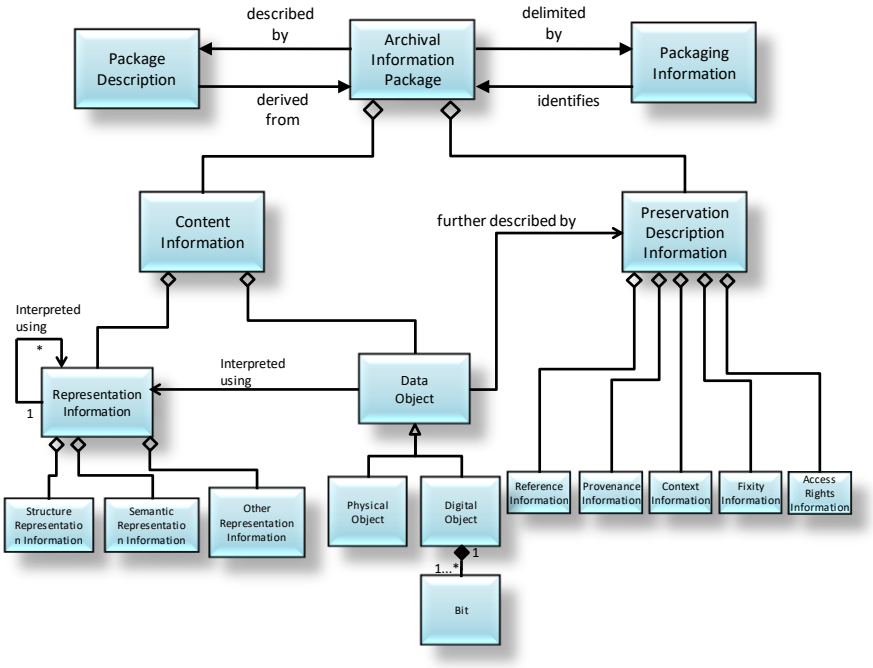


Figure 3-3 Detailed view of OAIS Archival Information Package (AIP)

Commentato [MWK38]: Should reference the OAIS-RM figure?

Commentato [DG39R38]: Could do.

Specific scenarios are listed next. In the following, for generality the term "User" can refer to a user, an archive (whether OAIS conformant or not) or an Information Creator or an Information Producer. The term "identifier" is used as a general term for something used to obtain the item required.

3.1 INFORMATION PRODUCER SENDS INFORMATION TO AN OAIS CONFORMANT ARCHIVE

Commentato [MWK40]: Bookmark for Mike Kearney – current farthest read.

Information is created by an Information Creator. It is sent to an OAIS Archive, via an Information Producer, which may be different from the Information Creator. OAIS defines, in general terms, a Submission Agreement that specifies the intended formats and content descriptions, and any other arrangements needed (such as delivery or transmission protocols), for the Data Submission Sessions. Such specifications establish the intended deliverables from the Producer and how they are represented on each delivery through physical media or in a telecommunication dialogue.

The Submission Agreement may be a document to be interpreted by a human. However, the extent to which these things can be computer interpretable is of interest for the OAIS-IF requirements.

The Information Producer may use the PAIS standard to encode the SIPs, in which case OAIS-IF may be used to define how the SIPs are transferred to the OAIS.

However, the Information Producer may use a different encoding for the SIPs, in which case OAIS-IF may be used to ensure that the SIPs may be understood by the OAIS.

Issues to be addressed include several which are mentioned as part of the Submission Agreement, but here the aim is to be more specific:

- How to arrange and carry out the transmission
- How to ensure that the Representation Information (Packaging Information) of the package is provided to the archive.
    - If the Producer uses PAIS then the PAIS standard itself can be referred to, but if in the future there are multiple versions of PAIS then the correct version must be identified.
- How to ensure that the Archive can extract components from the Information Package Data Object.
- The OAIS archive will expect to receive, perhaps over the course of several packages, the information required to create an AIP.

If a sequence of transmissions is to be arranged, with authentication mechanisms and fixity checks a context may need to be maintained for the Information Packages, for example to ensure packages are kept in sequence. Alternatively, the information may be provided in documentary form, rather than through an API.

3.2 INFORMATION PRODUCER SENDS INFORMATION TO A NON-OAIS CONFORMANT ARCHIVE

This scenario is similar to the one with the OAIS archive. Significant differences are that there might not be a formal Submission Agreement and the archive might not require all the components of an AIP.

3.3 OAIS CONFORMANT ARCHIVE SENDS INFORMATION TO AN INFORMATION USER

In this scenario the user searches for the information he/she requires and the OAIS Archive sends back information as a response to the search and then the information requested in the form of one or more DIPs.

The following steps are involved:

1. The user inputs search criteria and identifies repositories which contain required information.
2. The user chooses one (or more repositories) and queries what is available.
    a. The user may need to log in to see what he is allowed to access.
3. The user then obtains the information from the repository.
    a. The user may simply get a copy of the whole AIP or

b. The user may get a piece of information created by the repository suitably processing its holdings.

OAIS-IF treats queries and responses as messages in Information Packages. The DIPs are also Information Packages.

Since the archive is OAIS conformant then according to one of the Mandatory Responsibilities a user who is a member of the Designated Community must be able to

1) obtain enough Representation Information to understand/use the Data Object in the DIP and

2) obtain all the information related to Authenticity, and so the OAIS-IF must support this.

If the user is not a member of the Designated Community, then it does no harm to allow him/her to have the same functionality but in this case the Archive will not guarantee to be able to supply enough Representation Information to allow that user to understand/use the Data Object. This would be the case if the provided technical lexicon or jargon is not familiar to a consumer from outside the Designated Community.

3.3.1 AN INFORMATION USER WANTS TO OBTAIN INFORMATION FROM A ARCHIVE (OAIS COMPLIANT OR NOT)

1. User uses an "identifier" for repository to search and select specific information to obtain an object identifier.
2. The user requests the object or part of an object.
3. The user receives the object requested.

3.3.2 IF REQUIRED, INFORMATION USERS ARE AUTHENTICATED AND AUTHORIZED

1. User requests authentication
2. The user provides the appropriate username/password or private key etc

3.3.3 AN INFORMATION USER WISHES TO GET AN AIP

1. The object identifier is confirmed as pointing to an AIP rather than any other object
2. The Data Object of the AIP is retrieved
3. The Representation Information, of which Packaging Information is a part, associated with the Data Object is retrieved.
4. The Package Description Information is retrieved if required.

3.3.4 AN INFORMATION USER WISHES TO GET INFORMATION DERIVED FROM AN AIP

1. The user uses the identifiers for one or more AIPs to obtain object identifiers for the components of the AIPs.
2. The user may request some operation to be performed on the components to create new pieces of Information.
3. The user uses that identifier to obtain identifiers for any components of that component for example if the original AIP is an AIC then identifiers for the component.

AIPs can be obtained and then identifiers for the components of those AIPs can be obtained, and so on.

3.3.5 INFORMATION IS TRANSFERRED AS ONE OR MORE INFORMATION PACKAGES

1. The user uses the object identifier that has been used to obtain local identifiers for the various components of the AIP, or other pieces of information.
2. It constructs the object to be transferred e.g.
    a. Extracting the components from internal storage such as a database or filestore
    b. Some components may have sub-components such as individual events relevant to Provenance

3.4 NON-OAIS CONFORMANT ARCHIVE SENDS INFORMATION TO AN INFORMATION USER

This scenario is similar to the previous one except that the archive may not be able to guarantee to be able to supply Representation Information or evidence about Authenticity.

3.5 OAIS CONFORMANT ARCHIVE EXCHANGES INFORMATION TO ANOTHER OAIS ARCHIVE

The previous scenarios involving an OAIS could apply here, with one of the OAISes acting as either Producer or User. The special case is where the first OAIS should be able to send a copy of a complete AIP to the second OAIS, either as a single Information Package or as several related packages.

3.6 OAIS CONFORMANT ARCHIVE EXCHANGES INFORMATION WITH A NON-OAIS ARCHIVE

In this case either archive could play the role of Producer or User. The difference from the previous scenario is that complete copies of AIPs are unlikely to be sent from the non-OAIS conformant archive.

3.7 NON-OAIS CONFORMANT ARCHIVE EXCHANGES INFORMATION WITH ANOTHER NON-OAIS ARCHIVE

This scenario is similar to the previous one but, since non-OAIS archives may not actually have AIPs, copies of complete AIPs are not likely to be exchanged.

3.8 INFORMATION CREATOR SENDS INFORMATION TO INFORMATION PRODUCER

Information is created by an Information Creator. This may involve obtaining Information from other sources, where the Information Creator acts as an Information User.

It is sent to an Information Producer, which may be different from the Information Creator.

3.9 EARLIER USE CASE WORK

See https://www.dropbox.com/s/zx3gl35dq1vqgf6/UseCases_jsh_190110.xlsx?dl=0 , https://www.dropbox.com/s/r3jf3ticr86vgx3/UseCase_Service_Mapping_191014.xlsx?dl=0 and https://www.dropbox.com/s/59787kz3ttcwrl8/OAIS-IF%20Use%20Cases%2020181105.docx?dl=0 for earlier work on use cases.

4 OAIS-IF REQUIREMENTS

The term “archive” is used for a general information repository, whether conformant to OAIS or not; the term “OAIS” is used where the archive is OAIS conformant.

The following (incomplete) list of requirements for APIs have been identified so far. Requirements are not placed on the internal workings of the various entities, however some requirements may imply functionality from the archive, but if this functionality is not available then a NULL response may be made, and will be acceptable.

Table 1 Requirements for OAIS-IF - TO BE COMPLETED

REQUIREMENTS	SCENARIO
Req 1) APIs, based on, but not restricted to, the OAIS Information Model will be available.	3.1, 3.3, 3.5, 3.3.1, 3.3.3, 3.3.5
Req 2) An API which allows a negotiation to allow additional Representation Information to be provided, if available, should be part of the set of APIs.	3.3
Req 3) An API which allows a negotiation on transformations of the Data Object before transmission, should be part of the set of APIs.	3.3.4
Req 4) An API which allows the parties to agree on or discover a communications protocol to use should be part of the set of APIs.	3.1
Req 5) An API to allow the archive to be able to verify that a user requesting access to the archive is authorized should be part of the set of APIs.	3.3.2
Req 6) An API to allow the archive to be able to accept SIPs should be part of the set of APIs, in particular	3.1
Req 6-1) Allow the archive to verify that Producer requesting access to the archive is authorized.	3.1
Req 6-2) Make available the definitions of the types of submission packages that the archive will accept.	3.1
Req 6-3) one or more interfaces which can be used to submit an SIP and reports back on status of the ingest	3.1
Req 7) APIs to support search of the archive to provide an identifier for required information should be part of the set of APIs.	3.3
Req 8) APIs to retrieve information, as Information Packages, given one or more identifiers should be part of the set of APIs.	3.3.1

5 OAIS-IF CONCEPTS AND RATIONALE

This subsection provides several diagrams to illustrate the concept of the OAIS-IF architecture.

A simple architecture layer diagram based on Figure 2-3 showing a client connecting to an archive expanded to show how examples of existing software would be used in this case.

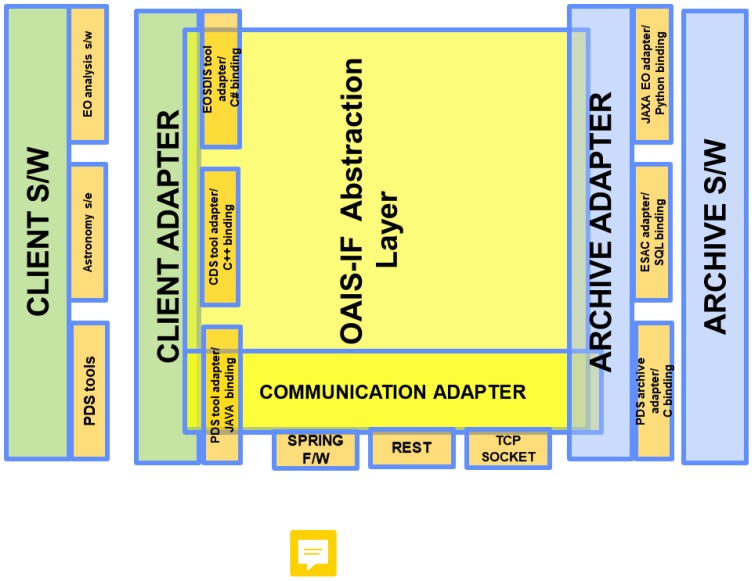


Figure 5-1 provides a depiction of the relationship of OAIS-IF components showing a Producer, Archive and Consumer, akin to the OAIS Functional Model. It is a layered architecture. It also "maps" from the document tree to the components of the architecture.

Figure 5-1: Relationship of OAIS-IF components in a layered architecture.

The overall flow of the figure from top to bottom is:

- Application interfaces (Producer and Consumer) at the top.
- An Archive Abstraction Layer (AAL) in the middle (similar to the MAL from the SM&C WG). As in the MAL and other abstraction layers, the AAL function is to hid the implementation details (such as archive specifications) allowing separation of concerns (such as user interfaces) to facilitate interoperability and platform independence. Our initial thoughts are that the functionality and interactions that pass through the AAL would include methods to (for example):
    - Allow Information objects, including Representation Information and other Objects to be sent and then used, for example for creating other pieces of Information or Information Packages and, in particular for creating Archival Information Packages (AIPs), the key component for long term preservation.
    - Allow a consumer to query an archive, or other information source, about Information available.
    - Allow access to a chosen Information.
    - Send Information from the information source to the user.

- Adapters or bindings on the bottom layer provide interfaces to specific types of archives or data formats.

Details of the other documents are provided in section 6.

The resulting architectural concept has the following characteristics:

- The overall layered Service Oriented Architecture (SOA) will provide modularity, portability, and the other well-known benefits of layered architectures.
- Only the standards for bindings required for spaceflight mission archives will be developed with the participation of the CCSDS space agencies. Other bindings can be developed elsewhere in the industry. For example, the Web/HTML adapters (or components of adapters) may be developed by a web archiving consortium, but will be available to will be available to others for their web preservation purposes.
- Four example bindings are illustrated above, but we anticipate that many other organizations would develop bindings for their specific data types or communities.
- The users (e.g. producers and consumers) have a standardized interface that is stable, and they have the ability to "plug in" adapters and bindings for many different data types to access data in other communities besides their own.
- Whereas currently archives usually provide interfaces only accessible to their Community, this approach will allow access to any authorized entity/system that has that adapter./binding. This should be a major advantage for stimulating cross-discipline research, which has been shown to be a significant contributor to advancement in technology.

Note that there may be name changes to the documents, and there may (temporarily) be inconsistencies in the set of documents defined.

Figure 5-2 illustrates the relationship of the documents relative to the flow of information from the producers, to the archives, to the consumers.

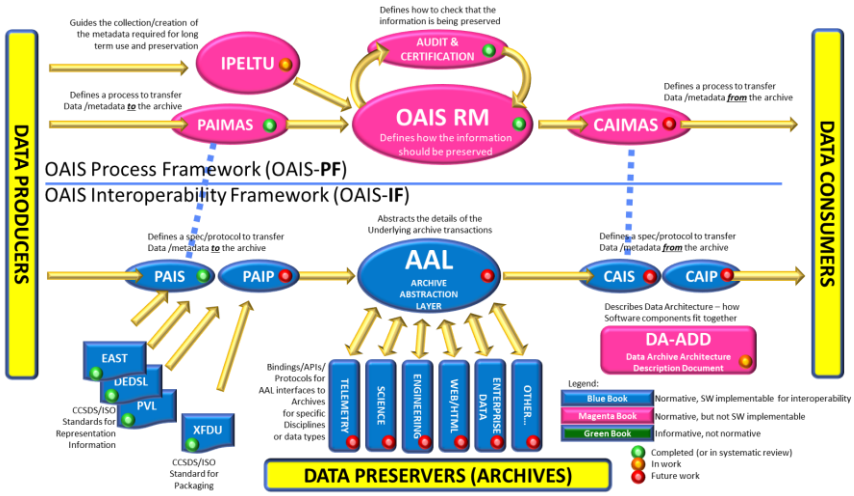


Figure 5-2 Relationship of the OAIS-IF documents to the principal actors (Producers, Archives and Consumers.)

6 OAIS-IF DOCUMENT STRUCTURE

The following subsections outline documents in the OAIS-IF set, with notes on their contents.

> **Note: The following diagrams are drafts and will be updated in due course** – see https://www.dropbox.com/s/c3ribtzpyz6ynh9/OAIS-IF-diagrams.pptx?dl=0 (Shared Dropbox: \CCSDS-DAI-Shared\Draft Documents\OAIS-IF Architecture Description\ OAIS-IF-diagrams.pptx)

6.1 OAIS DOCUMENT TREE

The following figure describes the OAIS Document Tree. The box colors follow the book color conventions of CCSDS: Magenta, Green and Blue.

See Figure 6-1. Particularly note the division between documents in the OAIS Process Framework (OAIS-PF)_and the OAIS Interoperability Framework (OAIS-IF). This document provides rationale, scenarios and requirements for the OAIS-IF. The OAIS-PF is shown as reference only. Acronyms are listed in section 1.6.1.

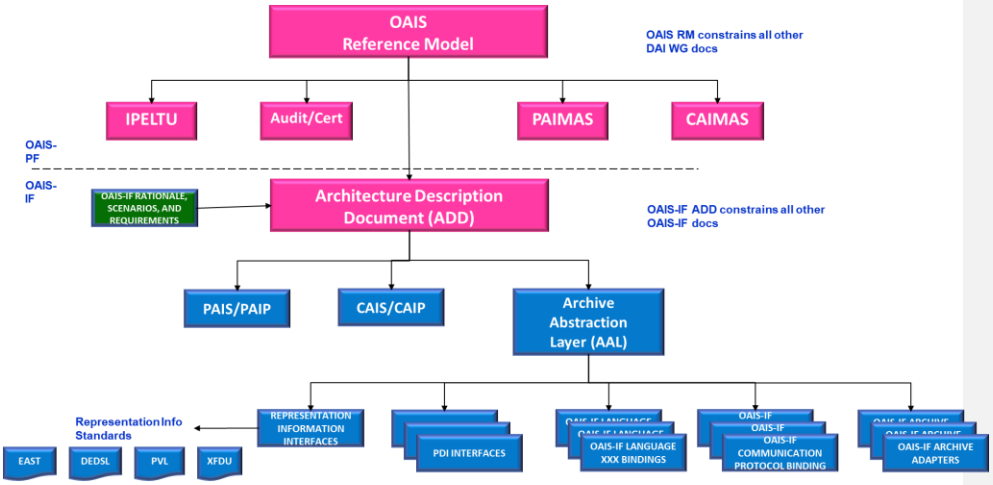


Figure 6-1: OAIS Document Tree

6.2 OAIS-IF RATIONALE, SCENARIOS, AND REQUIREMENTS GREEN (THIS DOCUMENT)

- An overview of the architecture, what problems we believe the set of standards is to address, what it will cover and what it will not cover (we had lots of discussion of that) and how it relates to OAIS and the other DAI (and other) standards.
- What people/systems will be able to do which they cannot do now
- Some simple use cases to illustrate what we are addressing.
- Make it clear that we solve the central problem of how to deal with "bags of bits" in a general way applicable throughout the architecture.
- An indication of what producers, users and archives will need to change to benefit from these standards.

6.3 ARCHITECTURE DESCRIPTION DOCUMENT (ADD) MAGENTA

- An overview of how we think things will actually fit together e.g., Steve's overall diagram plus some of the other UML diagrams (Class, Sequence, Component etc) which show different parts of the system in more detail.

See
https://www.dropbox.com/s/my91nnl1o8qplfo/02_White_Book_Recommended_Standard_OAIS-IF_Draft_191216%20v5.3%2006_ComparedCombined_04_05_201222b_Cleaned.docx?dl=0 and
https://docs.google.com/document/d/14V0wN6nEnG2MaSMmNClRzuDrTxcB0zjv0XNw2pdYDI8/edit#

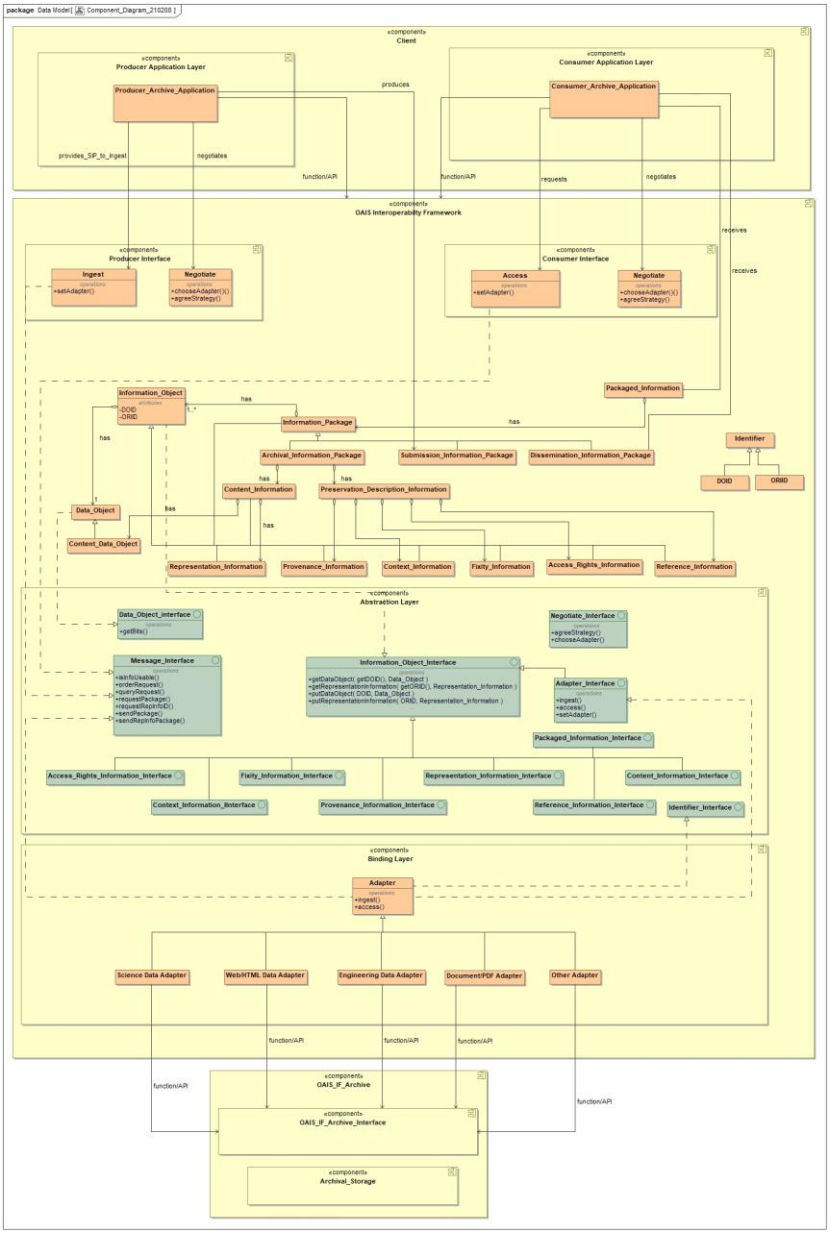


Figure 6-2 Draft Overview Diagram

6.4 ARCHIVE ABSTRACTION LAYER (AAL) BLUE

- The directly OAIS related Information Model interfaces, adding in methods and additional interfaces such as Identifiers etc.
- The OAIS Unique Selling Points of understandability
    - Negotiation interface and
    - a minimal interface for e Registry
- How we pass information between components e.g., Packaged Information
- Some specific interfaces for Representation Information, Provenance, Fixity, Access Rights, Reference (which may link to Identifiers) if we actually want interoperability. With modern programming languages we do not need to say that these are the only possible interfaces for these specific types of Information, but we recommend these.

See http://int-platform.digitalpreserve.info/wp-content/uploads/sites/5/2014/12/javadoc/model/ and http://int-platform.digitalpreserve.info/wp-content/uploads/sites/5/2014/12/javadoc/framework/ for an example of interfaces from an old project.

6.5 OAIS-IF ARCHIVE ADAPTERS MAGENTA

- What Archives need to have in order to interact – how to get from what the archives can provide to what the AAL demands?
- Simple examples of these
- What users need to have in order to be able to interact. This could be how to deal with encoded information in general terms plus some specific examples such as images, tables and trees.

6.6 OAIS-IF JAVA BINDING

- How to implement the AAL in Java (should be fairly straightforward)

6.7 OAIS-IF XXX BINDING

- How to implement the AAL in language XXX (difficulty depends on the language)

6.8 OAIS-IF COMMUNICATION PROTOCOL BINDING

- How to use existing communication protocols, including CCSDS, TCP/IP, SPRING, etc

6.9 PRODUCER-ARACHIVE INTEROPERABILITY PROTOCOL (PAIP) BLUE

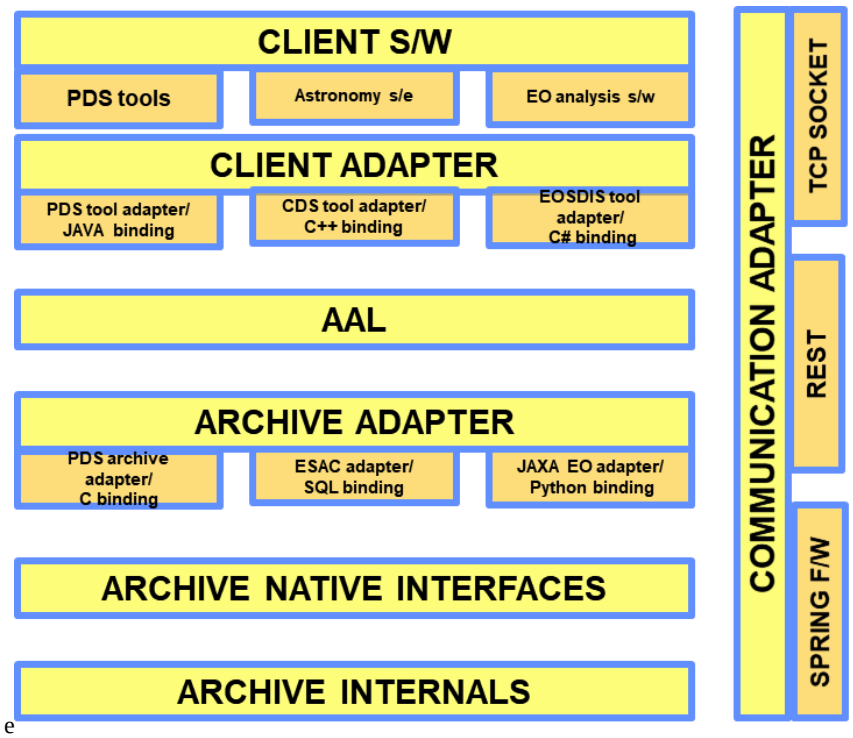
- SIPs and how to transfer them using:
    - PAIS
    - OAIS-IF documents above
- Notes on “out of band” communications to make arrangements.

6.10 CAIS/ PRODUCER-ARACHIVE INTEROPERABILITY PROTOCOL (CAIP) BLUE

- Queries – Use variety of existing standards
- DIPs

Figure 6-3 illustrates the use of the various components described by the documents described in this section, where the client software (e.g. PDS, Astronomy or EO software tools) uses an Adapter with a specific programming language binding to create objects specified by the AAL. The Archive adapter, with a specific programming language binding converts the AAL objects into ones which the Archive Native Interface can use to obtain the required information from the Archive Internals.

These objects are communicated through the Communications Adapter (shown here as being based on SPRING, REST or TCP, but others are possible). It is shown as a vertical bar to indicate that there are alternative ways of distributing the various pieces of software..



e

Figure 6-3 Illustrative software stack

7 OAIS-IF MORE DETAILED UML ANALYSIS

7.1 ASSUMPTIONS

1) We are focussed on information that comes out of and goes into things like OAISes and Applications (Producers/Consumers/Management).
2) We are concerned with OAIS functions within the archive **only** when they have characteristics that surface at user interfaces (users including other archives).
3) The information we are specially qualified to facilitate the exchange of is that which is related to OAIS Information Model.
4) The OAIS Information Model should form the basis of our API/protocol/binding/plugin, but things need to be added e.g. security.
5) We should not be concerned about optimisation right now.
6) We want to facilitate the exchange of information between disciplines and OAISes so it can be used and/or preserved.
7) An Information Package is a Data Object; its Packaging Information is its RepInfo which allows one to unpack the package and identify its components.. Together they make an Information Object.

The way to deal with an Information Object is the following

How to deal with an IO – fundamental ideas

a) Each Information Object (IO) must have an ID which identifies its RepInfo, an Object RepInfo ID (ORIID)
b) Given the ORIID the receiver of the IO can do one of the following:
   a. Use the IO because the receiver knows how to deal with things that have that ORIID
      OR
   b. Negotiate with the sender to see if it can send an alternative "form" which it does know how to use e.g. send a list of ORIIDs which it knows how to deal with and see if the sender can transform the Data Object to a preferred ORIID.
      1. If the negotiation is successful then (a) applies **OR**
      2. Use the ORIID to get as much of the RIN (RepInfo Network) from a registry as it needs to use the DO.
         a. The receiver may negotiate with the registry to get RepInfo it can use e.g. is there some Java software that can use the Data Object in the IO?
            i. To help decide which piece of the RIN to use, the RIN must have some mechanism for indicating which role the next pieces of RepInfo play e.g. Semantic, Structure or Other.

This is used throughout the application of the OAIS-IF.

7.2 DESIGN IDEAS

We want to design the system in a way that makes implementations easier and more widely usable.

1. For example if we can design software in a way that we can re-use the same piece of software in lots of different places in the system then that is a simplification because we don't need to write so much software.
2. On the other hand we may realise that we need to change the software to deal with different circumstances e.g. we might want to use the system with different types of networks, or with different sources of information or different ways of putting information together, or using different programming languages. If we break the design into layers we stand a better chance of isolating these various dependencies so that, for example, we can replace the software that deals with one network (say TCP/IP) with that which deals with another network (such as some CCSDS protocols), without changing anything else in the system.
3. Another simplification is whether we can re-use some existing design/software which itself helps with (1) and/or (2)
4. Existing software, which is specific to an Archive, may be used to access information.

The ideas here are aimed at designing the system to be able to re-use (see (1)) some pieces of software to deal with understandability – which pops up everywhere unless what we send is very rigidly specified.

The CCSDS MAL speaks to (3) above, but there are lots of other possibilities. The use of the JAVA repository and registry API would also be in line with (3).

For point (2) there are lots of possibilities and design choices. The discussion about, say, an API for Provenance was to provide a proposal to address part of this. Different Producers, Consumers and Archives also strongly suggest that we try to define layers where they are obvious now which isolate the specifics of each while keeping the rest of the system unchanged. Other layers may be identified later, perhaps breaking an initial layer into sub-layers.

UML diagrams support the development of designs to support these ideas.

7.3 (ARBITRARY) DECISIONS

Send information in OAIS Information Packages

- This allows us to send any information, so we can re-use software in many places.
- If a package goes into an OAIS we can call it an SIP
- If a package comes out of an OAIS we can call it a DIP
- If the package goes to or comes from a place other than an OAIS, then we just call it a package.

- If a package has everything needed for preservation of the (content) Information then it can be called an AIP.
  - Whether or not it actually has everything needed for preservation depends upon the Designated Community of the AIP, as defined by the OAIS, hence we have to decide:
    1. The Package needs enough information about the Designated Community in order to decide whether there is enough RepInfo etc

    OR

    2. The specific OAIS can declare that this Package is an AIP e.g. by setting a flag
       - This is probably the easiest way to go

One very convenient type of RepInfo is software that can be used in an application. Here are some proposed standard APIs

For an Information Object, software should support an interface:
- getDataObject (returns a DataObject – perhaps as ID (DOID))
- getRepInfo (returns RepInfo – perhaps as ORIID)

For an Information Package, software should support an interface:
- - getPackagingInfo (returns a ORIID)

The PackagingInfo has the following interface:
- getDataObject (returns a DataObject – perhaps as ID (DOID))
- getRepInfo (returns RepInfo – perhaps as ORIID)
- getProvenanceInfo (return ProvenanceInfo – perhaps as DOID - or NULL if not there)
- getReferenceInfo (return ReferenceInfo – perhaps as DOID - or NULL if not there)
- getAccessRightsInfo (return AccessRightsInfo – perhaps as DOID - or NULL if not there)
- getContextInfo (return ContextInfo – perhaps as DOID - or NULL if not there)
- getFixityInfo (return FixityInfo – perhaps as DOID - or NULL if not there)
- isAIP(return true or false)

For each of these Info Objects we can provide a default API e.g.

ProvenanceInfo (see https://docs.google.com/document/d/1YCyhBZKRP7IWhArdI3MnwahjZY5K93UsDZq-cWApYeI/edit?usp=sharing)

Contex

- getContext (returns an Information Object which contains text which provides the context)

ReferenceInfo

- getURI()

AccessInfo

Get????

FixityInfo

- getFixity(????)

RepInfo

7.4 UML DIAGRAMS

UML defines a number of different diagram types, each designed to show particular aspects of the system as any one particular diagram cannot show everything.

The following diagrams are similar, but not identical, to the text above.

7.4.1 USE CASES

The Use Cases shown in the diagram focus on the interactions between Consumers and Archives. The Producers are largely ignored here because

1. The Producer-Archive interactions are more complex
2. There are typically many fewer Producers than Consumers for any specific Archive,
3. Typically Producers would interact with a small number of Archives whereas Consumers would tend to interact with a large number of Archives.

Consumers will interact with familiar Archives, getting Information with which they are familiar (although this may not apply to Provenance Info etc). We show here that such Consumers use the Native Access Methods provided by the Archive. These Consumers will already know how to use the Information held by the Archive, and indeed they may be members of the Designated Community.

What we wish to focus on here are the interactions with other Consumers, those for which the Archive is unfamiliar. In this case we will need to provide extra Representation Information.

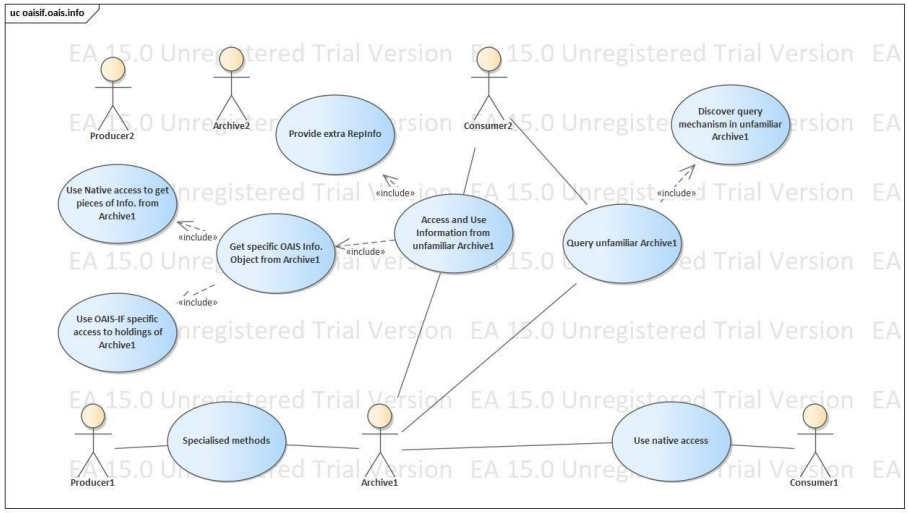


Figure 7-1 Use Case diagram
(click here for the file)

7.4.2 CLASS DIAGRAM - INTERFACES

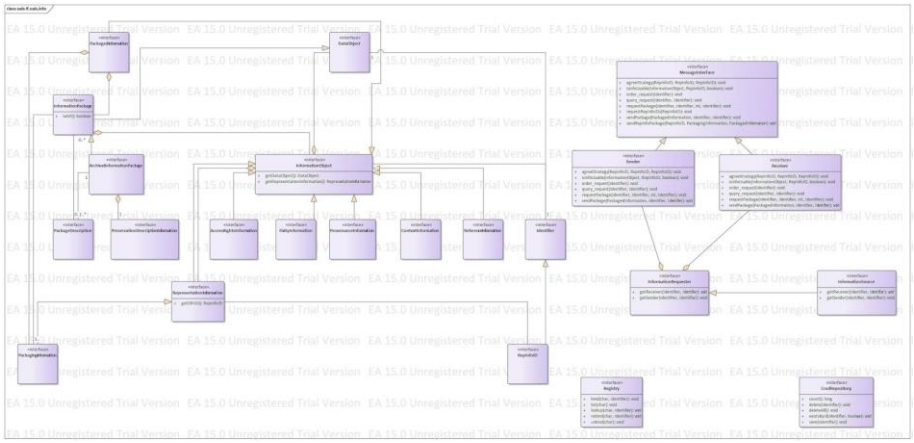


Figure 7-2 Class diagram

(click here for image)

This diagram shows the UML Class diagram, which uses only INTERFACES which need to be implemented rather than the Classes which implement them. These interfaces will be implemented using whatever language is required and built on whatever existing software libraries are available. These interfaces should form the basis of the PLUGIN implementation.

In addition to the OAIS Information Model there are:

- PackagedInformation - the InformationPackage plus its PackagingInformation
- Identifier - a general identifier, for example could be a DOI
- MessageInterface - at the moment this is missing the methods in Steve's diagram. The methods shown had previously been split into
- InformationRequester - something that requests information specified by an Identifier.
- InformationSender - something which sends information in the form of PackagedInformation
- CrudRepository - methods based on Java Interface (CRUD = create, Read, Update, Delete)
- Registry - methods based on Java Interface

7.4.3 SEQUENCE DIAGRAM

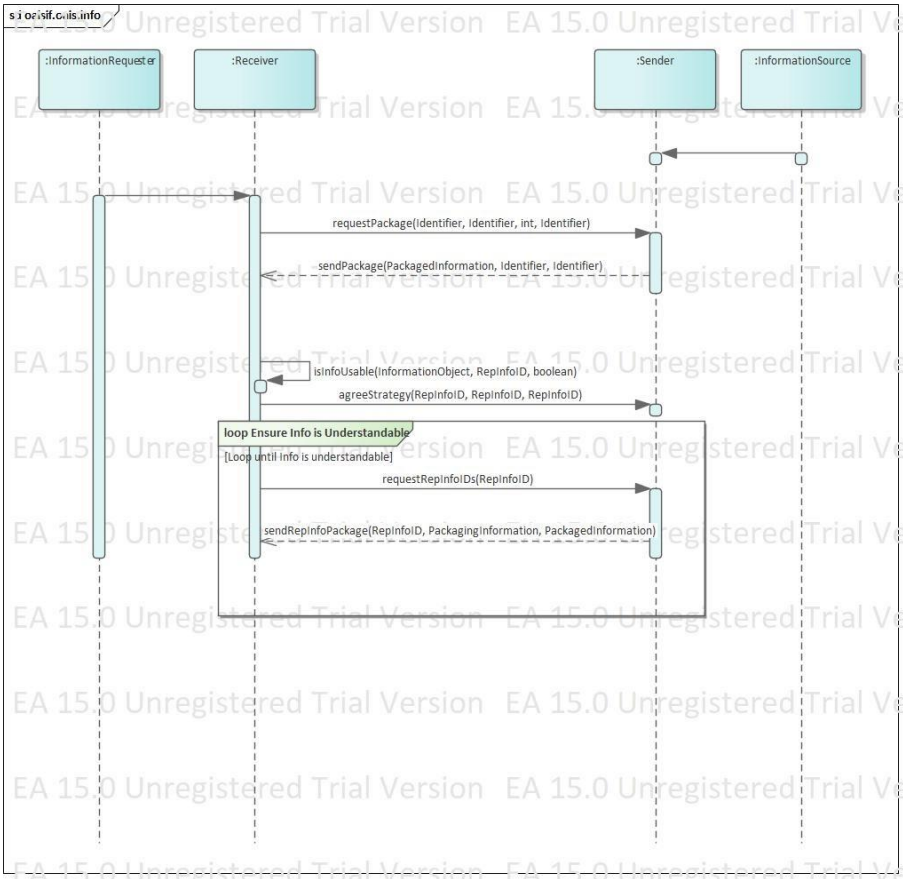


Figure 7-3 Sequence diagram
(click here for file)

The Sequence diagram shows the InformationRequester asks for some information for which (by some mechanism not covered here) it has an identifier.

On receipt of this request the InformationSender creates an InformationPackage, and then creates a PackagedInformation object by combining that with its PackagingInformation.

On receipt of the PackagedInformation the Requester decides whether or not it has enough RepInfo to understand/use the data.

If it has enough RepInfo then there is no need for further requests.

If on the other hand there is not enough RepInfo then

1. The Sender and Receiver agree on a Strategy.

2. The list of identifiers of RepInfo which the Requester "understands" is sent to the Sender.

3. The Sender then sends a PackagedInformation object which contains the appropriate RepInfo.

7.4.4 COMPONENT DIAGRAM

Component diagrams can be used to model the logical components that make up a system. They can be used to model the applications of an organization including their Provided and Required Interfaces and the information that is exchanged between the interfaces.

The diagram below shows the components showing that the Consumer interacts with the Consumer Abstraction Layer (CAL). The CAL uses the Message Interface to query for and request information from the Producer, at the Producer Abstraction Layer (PAL).

There is a negotiation, shown in the Sequence Diagram (which needs to be updated), to decide on whether, in order for the Consumer to be able to use the information returned, the PAL can either:

1. Send additional RepInfo to extend the RepInfo Network so that the Consumer will have enough RepInfo. The Producer may use a Registry/Repository of Representation Information to supplement the RepInfo it already has.

2. The Data Object could be Transformed to match what the Consumer can deal with.

To make the decision the PAL uses an Interoperability Table shown in the diagram (see accompanying Note) in order to decide on which strategy to use. It may be that both option (1) and option (2) are possible, in which case the Cost which is included in the Table will be used to decide.

The PAL and CAL together make up the Abstraction Layer in the PPT diagram.

To implement this, CCSDS (or one or more agencies) can provide Reference Implementations. A Domain Archive could initially simply use the Reference Implementation. It would only need to create the Interoperability Table, and supply the Applications (if any) to perform the Transformation, which should be chosen based on maintaining the Transformational Information Properties.

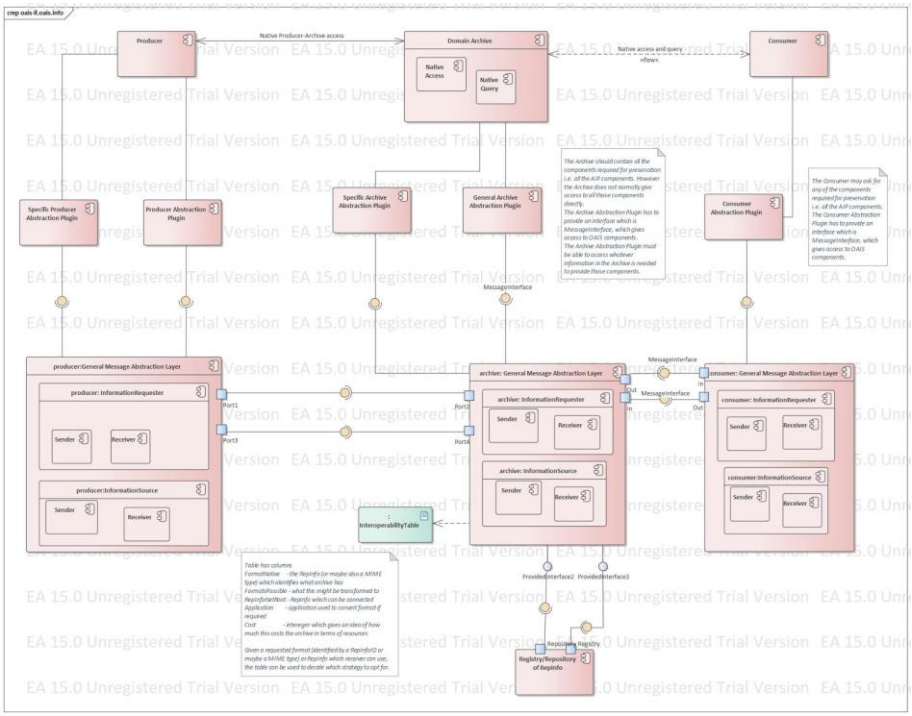


Figure 7-4 Component diagram
(click here for file)

7.4.5 DEPLOYMENT DIAGRAM

The Deployment diagram indicates (some of the) the software which runs on the physical nodes (e.g. computers).

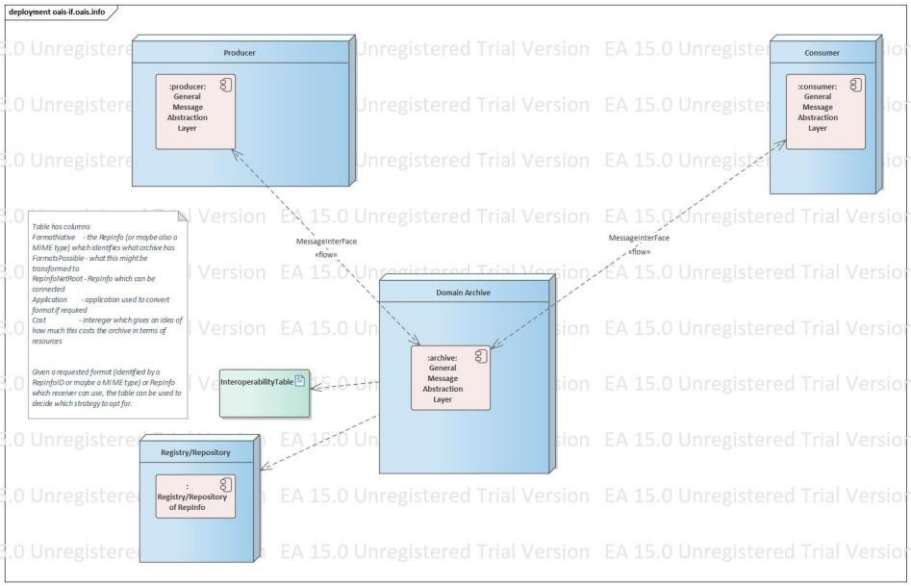


Figure 7-5 Deployment diagram
(click here for file)

7.5 TODO:

- QUERY - could use the Java Query Interface
  https://docs.oracle.com/javaee/7/api/javax/persistence/Query.html

This is not quite consistent with the CCSDS Message Abstraction Layer (see https://public.ccsds.org/Pubs/521x0b2e1.pdf), but if this seems a good way to go it may be relatively straightforward to do that. We then also may be able to use the MAL JAVA API (see https://public.ccsds.org/Pubs/523x1m1.pdf). A JAVA prototype implementation could be quite straightforward.

The binding to the MAL HTTP Transport (see https://public.ccsds.org/Pubs/524x3b1.pdf) may also be useful.

8 MORE DETAILED INTERFACES

8.1 REPRESENTATION INFORMATION INTERFACES

The Negotiation step which is described in the document defined in section 6.4 is to ensure that enough Representation Information is provided. This therefore means that the system should be able to recognize whether or not the is enough of the various types of Representation Information, and how to get more of the Representation Information Network if required. The Representation Information Interface document defines the required interfaces.

The initial idea is to collect summaries of approaches to encoding Representation Information (RI) and then try to extract commonalities, following this a general interface will be drafted.

8.1.1 OAIS DEFINITIONS

Representation Information*: The information that maps a Data Object into more meaningful concepts so that the Data Object may be understood in ways exemplified by Preservation Objectives.*

> *NOTE: An example of Representation Information for a bit sequence which is a FITS file might consist of the FITS standard which defines the format plus a dictionary which defines the meaning in the file of keywords which are not part of the standard. This would then allow the information in the FITS file to be used by a computer program to display the image which may be contained in the FITS file, together with the associated coordinate system so that a human can identify objects of interest, for example stars or galaxies. Alternatively, the computer program may identify such objects automatically.*

Representation Information Network*: The set of Representation Information that fully describes the meaning of a Data Object. Representation Information in digital forms needs additional Representation Information so its digital forms can be understood over the Long Term.*

Semantic Information: *The Representation Information that further describes the meaning of the Data Object beyond that provided by the Structure Information.*

Structure Information*: The Representation Information that imparts meaning about how the Data Object is organized.*

> *NOTE: For example, Structure Information maps bit streams to common computer types such as characters, numbers, and pixels and aggregations of those types such as character strings and arrays.*

Other Representation Information: *Representation Information which cannot easily be classified as Semantic or Structural.*

NOTE: For example, software, algorithms, encryption, written instructions and many other things may be needed to understand the Content Data Object in ways exemplified by the Preservation Objectives, all of which therefore would be, by definition, Representation Information, yet would not obviously be either Structure or Semantics. Information defining how the Structure and the Semantic Information relate to each other, or software needed to process a database file would also be regarded as Other Representation Information.

8.1.2 POSSIBLE REPINFO USE CASES

1. Do something with a Data Object (DO) using RI e.g. as specified in the Preservation Objectives
    1. Identify the RI associated with the DO
    2. Use Semantic Information to identify an information element required
        1. Identify the Semantic Info components of the RI related directly to the DO (note that much of the RIN will relate to pieces of RI within the RIN)
        2. List the Semantic Info components of the RI related to the DO
        3. Are the Semantic Info components things which is understandable/usable by the user
    3. Use Structure Information to extract the specific information element
        1. Identify the Structure Info components of the RI related directly to the DO (note that much of the RIN will relate to pieces of RI within the RIN)
        2. List the Structure Info components of the RI related to the DO
        3. Are the Structure Info components things which is understandable/usable by the user
    4. Use Other RI
        1. Identify the Other RI components of the RI related directly to the DO (note that much of the RIN will relate to pieces of RI within the RIN)
        2. List the Other RI components of the RI related to the DO
        3. Are the Other RI components things which is understandable/usable by the user
    5. In addition one can see that there are other considerations:
        - There may be equivalent “versions” of an example of Representation Information e.g. software which may be written in C, C++ Java etc - any of these may be used i.e. it may be (C-version OR Java version **OR** C++ version)
        - Some types of Representation Information may be required e.g.
            - Structure Representation Information e.g a CSV text file **AND**
            - Semantic Representation Information e.g. a dictionary with the meaning, units etc of the table columns

8.1.3 POSSIBLE REPINFO INTERFACE REQUIREMENTS

An Information Package (IP) will logically contain an Information Object (IO) which consists of the DO and RI. This would allow a user to identify the RI of the DO.

Presumably there are many possible implementations of the IP, and once ingested the components of the IP will be stored in some way in the user's systems.
The implementation details should be hidden by the Interoperability Interface (and associated API(s))
One possibility is for the RI to have as part of it some kind of "manifest" that allows one to cope with Use Cases like 1.2.1, 1.2.2, 1.3.1, 1.3.2, 1.4.1 and 1.4.2.

Also http://int-platform.digitalpreserve.info/wp-content/uploads/sites/5/2014/12/javadoc/model/info/digitalpreserve/interfaces/RepresentationInformation.html

8.2 PROVENANCE INFORMATION INTERFACES

The initial idea is to collect summaries of approached to encoding Provenance and then try to extract commonalities, following this a general interface will be drafted.

OAIS has the following definition:

Provenance Information*: The information that documents the history of the Content Data Object. This information tells the origin or source of the Content Data Object, any changes that may have taken place since it was originated, and who has had custody of it since it was originated. The Archive is responsible for creating and preserving Provenance Information from the point of Ingest; however, earlier Provenance Information should be provided by the Producer. Provenance Information adds to the evidence to support Authenticity.*
Relevant approaches:

- Open Provenance Model https://openprovenance.org/opm/
- PREMIS Data Model

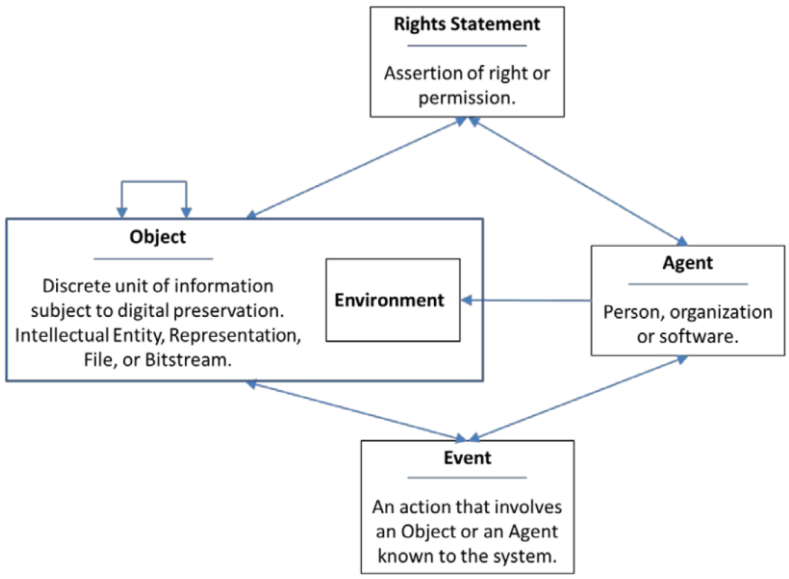


Figure 8-1 PREMIS Model

Object, Environment, Rights, Agent, Event

- PROV (W3C) https://www.w3.org/TR/prov-overview/

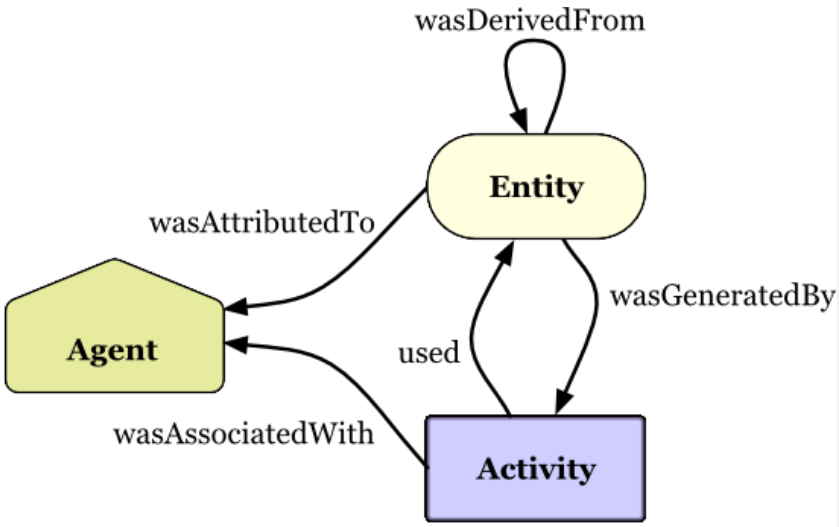


Figure 8-2 PROV Model

Entity, Agent, Activity

- IPAPI - https://www.cl.cam.ac.uk/~acr31/pubs/carata-ipapi.pdf
- Taverna Provenance API - http://www.taverna.org.uk/api/net/sf/taverna/t2/provenance/api/ProvenanceAccess.html

8.2.1 POSSIBLE PROVENANCE USE CASES

1. Use Provenance Information to find out what has happened to a DO
    1. Identify the Provenance Information associated with a Data Object (DO)
    2. List provenance events in time order
    3. For each event, obtain (e.g. following PREMIS model): Object, Environment, Rights, Agent, Event - more specifically
        1. Data Object Identification Information (Reference Information??)
        2. Information about systems used
        3. Access Rights Information
        4. Agent Identification Information
        5. Event Description Information

8.2.2 POSSIBLE PROVENANCE INTERFACE REQUIREMENTS

Each of the types of Information identified above will be made up of DO and RI. Representation Information Concept apply.

Also see http://int-platform.digitalpreserve.info/wp-content/uploads/sites/5/2014/12/javadoc/model/info/digitalpreserve/interfaces/ProvenanceInformation.html

8.3 FIXITY INFORMATION INTERFACES

TBD

See http://int-platform.digitalpreserve.info/wp-content/uploads/sites/5/2014/12/javadoc/model/info/digitalpreserve/interfaces/FixityInformation.html

9 THOUGHTS ON OAIS-IF DESIGN AND IMPLEMENTATION

9.1 PURPOSE

At the meeting on Tuesday the question arose as to what the (minimum) responsibilities of a repository would be to be able to use OAIS-IF.

This note, which is an introduction to [3], presents a high level view of an approach to the design of OAIS-IF which will answer this question. If this approach is accepted then we can look at the details of how this could be taken further, perhaps by taking [3] forward.

Therefore, a reviewer of this note should not expect to get all the answers at this point. Only if/when this approach is accepted, or at least thought to be potentially useful, will the next stage of analysis be undertaken.

9.2 BACKGROUND

Archives have, and will have, different capabilities.

Some repositories will not be able, for one reason or another, to be compliant with the OAIS-IF requirements.

What is the minimum that a repository must be able to do i.e., what cannot be done within the framework but that the framework needs in order to “fill in the gaps”?

Given that we are basing this OAIS-IF approach on the OAIS Reference Model then one would think that the Framework needs to know:

How do I (the Framework) get the data object?

How do I get its Representation Information?

How do I get its Provenance Information?

How do I get its Access Rights Information?

How do I get its Fixity Information?

How do I get its Context Information?

How do I get its Reference Information?

9.3 ANALYSIS

These questions are of two types:

How to get a Data Object

How to get an Information Object, which involves getting:

a. The Data Object of that Information Object (see point 1) plus

b. The Representation Information of that Information Object (go back to the start of point 2)

Going back to the questions one might reasonably imagine then the range of answers could include:

- Use this URL
- Use this identifier in a call to Glacier
- Use this TCP/IP socket and use this protocol
- Use ABC (a generic name used below)
- Etc

Of course the answer may be "I have no way for you to get these things" e.g. if the archive does not keep Provenance. In programming terms this would be equivalent to returning a NULL. We would then need to decide whether or not to accept NULL. Or perhaps we could accept NULL in some cases e.g. OK not to have CONTEXT, but must have PROVENANCE. This might be regarded as a PROFILE for interoperability for an archive, which one finds in some communication standards.

Bear in mind that Provenance Information, for example, could be as simple as an ASCII English text file which says "I do not have any Provenance". Until we have a way to evaluate the Provenance (in this example) then this could be regarded as the default answer if we receive NULL, and would not be any different from the archive creating this trivial text message and returning it to the OAIS-IF instead of NULL.

9.4 HOLLYWOOD PRINCIPLE

One way to simplify the OAIS-IF design would be to use the Hollywood Principle i.e. "don't call us, we'll call you".

In programming design pattern terms this could be an Inversion of Control Pattern or perhaps an Observer Pattern.

As an example, to answer the question "How do I get the Provenance Information for the Data Object which you refer to as (let's say) DOID?" the steps would be:

1. I (the archive) refer to the Provenance Information Object as IOD1
2. Get the Information Object which is Provenance Information as follows:
    a. Use the method ABC with parameters (X, Y, Z) to get the Data Object of the Provenance Information (e.g. a PREMIS file), which I refer to as DOID2, with identifier RIID to get its Representation Information
    b. Use method GHI with parameters (T,U,V) to get the Data Object of the Representation Information (e.g. a description of how to extract elements of Provenance such at event time), which I refer to as DOID3.
        i. Repeat to get as much Representation Information as required [2]
3. If the description allows one to implement a well defined Provenance Interface [1] then the Provenance elements can be accessed programmatically. Otherwise it may require human intervention.

9.5 REFERENCES

[1] A concept paper on Provenance – see https://docs.google.com/document/d/1YCyhBZKRP7IWhArdI3MnwahjZY5K93UsDZq-cWApYeI/edit?usp=sharing

[2] A concept paper on Representation Information - https://docs.google.com/document/d/1TcYIDKc9WMdyK1POSuitQjdWc5VtlL-YG8Qkt8c8V18/edit?usp=sharing

[3] A concept paper on OAIS-IF ideas – see https://docs.google.com/document/d/14V0wN6nEnG2MaSMmNClRzuDrTxcB0zjv0XNw2pdYDI8/edit#