Details of interfaces

Diagram

Description automatically generated with low confidence

Part 1 of interface diagram

Diagram

Description automatically generated with low confidence

Part 2 of interface diagram

Diagram

Description automatically generated with low confidence

Part 3 of interface diagram

Example documentation of Interfaces

The following are some examples of the documentation about the interfaces. Apologies

## Interface InformationObject

All Known Subinterfaces:

AccessRightsInformation, AndRepInfoGroup, ArchivalInformationPackage, ContextInformation, DescriptiveInformation, DisseminationInformationPackage, FixityInformation, InformationPackage, OrRepInfoGroup, OtherRepresentationInformation, PackagingInformation, PreservationDescriptionInformation, ProvenanceInformation, ReferenceInformation, RepInfoGroup, RepInfoLabel, RepInfoNetwork, RepresentationInformation, RepresentationInformationPackage, SemanticRepInfo, StructureRepInfo, SubmissionInformationPackage

### public interface InformationObject

The OAIS Information Object [OAIS section 4.2.1.1] is composed of a Data Object that is either physical or digital, and the Representation Information that allows for the full interpretation of the data into meaningful information. This model is valid for all the types of information in an OAIS. Note that here the InformationObject is simply a way of connecting the DataObject to its RepresentationInformation, it does not at this level of abstraction imply that the InformationObject is itself the "information" which is usable by the Designated Community, That "information" results from the instance-specific combination of the DataObject with its RepresentationInformation. The RepresentationInformation will normally be returned via a RepInfoID.

### Method Summary

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| DigitalObjectID | getDigitalObject() | Get the Identifier for the data object |
| Identifier | getIOID() | Get the Identifier for the Information object |
| RepInfoID | getRepresentationInformation() | Get the Identifier for the RepresentationInformation |
| void | setDigitalObject​(DigitalObject dataob) |  |
| void | setOID​(Identifier oid) | Set the ID which points to the RepInfo (e.g. |
| void | setRepresentationInformation​(RepresentationInformation ri) |  |

## Interface AccessRightsInformation

All Superinterfaces:

InformationObject

### public interface AccessRightsInformation extends InformationObject

**Access Rights Information**: The information that identifies the access restrictions pertaining to the Data Object, including the legal framework, licensing terms, and access control. It contains the access and distribution conditions stated within the Submission Agreement, related to both preservation (by the OAIS) and final usage (by the Consumer). It also includes the specifications for the application of rights enforcement measures.

### Method Summary

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| String | getRightsEncoding() | Description of the system under which the rights text were recorded |
| String | getRightsText() | Get the text of the rights definition |
| void | setRightsEncoding(String re) | Set the encoding of the rights definition |
| void | setRightsText(String rt) | Set the text of the rights definition |

### Methods inherited from interface info.oais.interfaces.InformationObject

getDigitalObject, getIOID, getRepresentationInformation, setDigitalObject, setOID, setRepresentationInformation

## Interface ArchivalInformationPackage

All Superinterfaces:

InformationObject, InformationPackage

All Known Subinterfaces:

RepresentationInformationPackage

### public interface ArchivalInformationPackage extends InformationPackage

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

### Method Summary

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| boolean | **isValid()** | Returns TRUE if the AIP is valid i.e. |

### Methods inherited from interface info.oais.interfaces.InformationObject

getDigitalObject, getIOID, getRepresentationInformation, setDigitalObject, setOID, setRepresentationInformation

### Methods inherited from interface info.oais.interfaces.InformationPackage

getInformationObject, getPackageDescription, getPackagingInformation, getPDI, getVersion, setInformationObject, setPackageDescription, setPackagingInformation

## Interface DigitalObject

### public interface DigitalObject

An OAIS Digital Object defined as an object composed of a set of bit sequences.

### Method Summary

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| byte[] | **getDataResource()** | Get access to the bytes. |
| void | **setDataResource​(byte[] dataResource)** |  |

## Interface InformationPackage

All Superinterfaces:

InformationObject

All Known Subinterfaces:

ArchivalInformationPackage, DisseminationInformationPackage, RepresentationInformationPackage, SubmissionInformationPackage

### public interface InformationPackage extends InformationObject

An OAIS Information Package is a container that contains two types of Information Objects, the Content Information and the Preservation Description Information (PDI); the Information Package can be associated with two other types of Information Objects, Packaging Information and Package Descriptions. InformationPackage implements the java.lang.Comparable interface which means that two InformationPackages may be compared and, where possible, the output from compareTo() will show whether one is a later version of the other.

### Method Summary

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| InformationObject | getInformationObject() | This gets the InformationObject in the package. |
| PackageDescription | getPackageDescription() | This gets the description of the package. |
| PackagingInformation | getPackagingInformation() | This gets the packaging information of the package. |
| Preservation  Description  Information | getPDI() | This gets the PDI in the package. |
| Version | getVersion() | An opaque object which may be used by the Comparable interface |
| void | setInformationObject​(InformationObject ci) | Set the InformationObject for the package |
| void | setPackageDescription​(PackageDescription pde) | Set the description of the package |
| void | setPackagingInformation​(PackagingInformation pi) | Set the packaging information for the package |

### Methods inherited from interface info.oais.interfaces.InformationObject

getDigitalObject, getIOID, getRepresentationInformation, setDigitalObject, setOID, setRepresentationInformation

## Interface ProvenanceInformation

All Superinterfaces:

InformationObject

### public interface ProvenanceInformation extends InformationObject

**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.

### Method Summary

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| ProvenanceStep[] | **getProvenanceSteps()** | Return an array of Provenance steps |

### Methods inherited from interface info.oais.interfaces.InformationObject

getDigitalObject, getIOID, getRepresentationInformation, setDigitalObject, setOID, setRepresentationInformation

## Interface ProvenanceStep

### public interface ProvenanceStep

A ProvenanceStep records one action (step) which has been performed (by an Actor) on the object at time specified by a TimeStamp. TODO: review this

### Method Summary

| **Modifier and Type** | **Method** | **Description** |
| --- | --- | --- |
| Actor | **getActor()** | The Actor who performed the action/step |
| Step | **getStep()** | The action or step which was carried out |
| Date | **getTimeStamp()** | The TimeStamp for the action/step |
| void | **setActor​(Actor actor)** | Set the actor responsible for the step |
| void | **setStep​(Step step)** | Set the action involved |
| void | **setTimeStamp​(Date time)** | Set the time at which the step took place. |