



Project no. 033572

## CASPAR

**C**ultural, **A**rtistic and **S**cientific knowledge for **P**reservation, **A**ccess and **R**etrieval

**Instrument:** Information Society Technologies

**Thematic Priority:** 2.5.10 Access to and preservation of cultural and scientific resources

# D2101:PROTOTYPE O AIS - INFRASTRUCTURE



---

|                      |                                 |
|----------------------|---------------------------------|
| Document identifier: | <b>CASPAR-D2101-RP-0101-1_0</b> |
| Submission Date:     | <b>26-05-2008</b>               |
| Due Date:            | <b>15-02-2008</b>               |
| Work package:        | <b>2100</b>                     |
| Partners:            | <b>All Partners</b>             |
| WP Lead Partner:     | <b>STFC</b>                     |
| Document status      | <b>FINAL</b>                    |

---

Abstract: This document provides a summary of the links between the CASPAR infrastructure, and the fundamentals of O AIS. Associated with it are draft reports of knowledge management architecture and tools.



**Delivery Type** Report  
**Author(s)** CASPAR Consortium

**Approval** David Giaretta

**Summary**

**Keyword List**

**Availability**  PUBLIC

### Document Status Sheet

| Issue | Date       | Comment                          | Author                           |
|-------|------------|----------------------------------|----------------------------------|
| 0_1   | 01-11-2007 | Initial draft                    | David Giaretta                   |
| 0_2   | 26-02-2008 | Revised draft                    | David Giaretta                   |
| 0_3   | 20-03-2008 | Additional text about interfaces | David Giaretta                   |
| 0_4   | 22-05-2008 | Text about Functional Model      | David Giaretta                   |
| 0_5   | 25-05-2008 | Some minor amendments            | Simon Lambert and David Giaretta |
| 1_0   | 25-05-08   | Final version                    | David Giaretta                   |
|       |            |                                  |                                  |





### Project information

|                        |   |
|------------------------|---|
| Project acronym:       | <b>CASPAR</b>   |
| Project full title:    | <b>Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval</b> |
| Proposal/Contract no.: | <b>IST-2006-033572</b>  |

### Project Officer: Carlos Oliveira

|          |   |
|----------|---|
| Address: | <p>INFO-E3<br/>Information Society and Media Directorate General<br/>Content - Learning and Cultural Heritage</p> <p>Postal mail:<br/>Bâtiment Jean Monnet (EUFO 1167)<br/>Rue Alcide De Gasperi / L-2920 Luxembourg</p> <p>Office address:<br/>EUROFORUM Building - EUFO 1167<br/>10, rue Robert Stumper / L-2557 Gasperich / Luxembourg</p> |
| Phone:   | +352 4301 33052   |
| Fax:     | +352 4301 33190   |
| Mobile:  |   |
| E-mail:  | Carlos.Oliveira@ec.europa.eu  |

### Project Co-ordinator: David Giarretta

|          |   |
|----------|---|
| Address: | STFC (formerly CCLRC), Rutherford Appleton Laboratory<br>Chilton, Didcot, Oxon OX11 0QX, UK |
| Phone:   | +44 1235 446235   |
| Fax:     | +44 1235 446362   |
| Mobile:  | +44 (0) 7770326304  |
| E-mail:  | <a href="mailto:d.l.giarretta@rl.ac.uk">d.l.giarretta@rl.ac.uk</a>                          |





## CONTENT

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUCTION.....</b>                                     | <b>5</b>  |
| 1.1      | PURPOSE OF THIS DOCUMENT.....                                | 5         |
| 1.2      | HOW TO READ THIS DOCUMENT.....                               | 5         |
| 1.3      | APPLICABLE DOCUMENTS AND REFERENCE DOCUMENTS .....           | 6         |
| 1.4      | GLOSSARY.....  | 6         |
| <b>2</b> | <b>OAIS MODELS.....</b>                                      | <b>8</b>  |
| <b>3</b> | <b>OAIS INFORMATION MODEL .....</b>                          | <b>9</b>  |
| 3.1      | PRESERVATION DESCRIPTION INFORMATION .....                   | 10        |
| 3.1.1    | <i>Fixity Information</i> .....                              | 11        |
| 3.1.2    | <i>Reference Information</i> .....                           | 12        |
| 3.1.3    | <i>Context Information</i> .....                             | 12        |
| 3.1.4    | <i>Provenance Information</i> .....                          | 12        |
| 3.2      | EXTENDING THE OAIS INFORMATION MODEL .....                   | 13        |
| 3.3      | OAIS INFORMATION MODEL PIM .....                             | 13        |
| 3.4      | OAIS INFORMATION MODEL INTERFACES .....                      | 15        |
| 3.4.1    | <i>Additional concepts</i> .....                             | 17        |
| 3.4.1.1  | <i>Identifier</i> .....                                      | 17        |
| 3.4.2    | <i>Messaging</i> .....                                       | 19        |
| 3.5      | PACKAGING.....   | 20        |
| 3.5.1    | <i>AIP</i> .....   | 22        |
| 3.5.2    | <i>API for packages</i> .....                                | 23        |
|          | INFORMATIONPACKAGE .....                                     | 23        |
| 3.6      | REPRESENTATION INFORMATION .....                             | 26        |
| 3.6.1    | <i>RepInfo toolbox</i> .....                                 | 26        |
| 3.7      | REGISTRY/REPOSITORY OF REPRESENTATION INFORMATION.....       | 26        |
| 3.7.1    | <i>Networks of Representation Information</i> .....          | 26        |
| 3.7.2    | <i>Knowledge Management</i> .....                            | 27        |
| 3.7.3    | <i>Obtaining additional Representation Information</i> ..... | 28        |
| <b>4</b> | <b>FUNCTIONAL MODEL .....</b>                                | <b>29</b> |
| 4.1      | PRESERVATION PLANNING .....                                  | 30        |
| 4.2      | DATA MANAGEMENT .....  | 30        |
| 4.3      | ARCHIVAL STORAGE .....                                       | 30        |
| 4.4      | ACCESS .....   | 30        |
| 4.5      | INGEST .....   | 30        |
| <b>5</b> | <b>CONCLUSIONS.....</b>                                      | <b>32</b> |





## 1 INTRODUCTION

### 1.1 PURPOSE OF THIS DOCUMENT

This document provides a summary of the links between the CASPAR infrastructure, and the fundamentals of OAIS. It shows how the developments within the CASPAR project are in line with OAIS; such an alignment is an essential part of CASPAR's approach to preservation. Associated with it are draft reports of knowledge management architecture and tools.

The infrastructure we describe is in line with the OAIS Reference Model [R2], the CASPAR Conceptual Model [D1201] and the CASPAR Architecture [D1301]

### 1.2 HOW TO READ THIS DOCUMENT

Several specific components of the CASPAR infrastructure are covered in other reports, in particular *CASPAR D2201 Preservation DataStore Interface* [A3], *CASPAR D2301 Report on OAIS Access Model* [A4] and *CASPAR D4102 Integrated report of R&D activities on the Cultural, Performing Arts and Science data Preservation Activities* [A5]. Such material will therefore only be touched upon here, with references provided to the more extensive coverage. This document should be read as laying the foundations for these components, establishing links between their functions and the OAIS models. One aspect to which particular attention is devoted is the distinction between what is domain-independent (generic) and what is domain-specific. This distinction is important in any practical preservation environment because it directly affects what may be reused and what must be constructed from scratch for each discipline. It therefore has implications for the take-up of the environment.

Section 2 briefly introduces the role of models in OAIS.

Section 3 focuses on the OAIS Information Model and shows how, in a fairly natural way, one can define some fundamental interfaces derives from this model, and use these as fundamental interfaces in components. This is a key step in moving from the generalities of OAIS to an implemented preservation environment.

In Section 4 the Functional Model components are linked to other aspects of the CASPAR infrastructure and including a discussion as to why some are not key components within CASPAR.





### 1.3 APPLICABLE DOCUMENTS AND REFERENCE DOCUMENTS

#### Applicable documents

- [A1] Description of Work, April 2006  
([http://www.casparpreserves.eu/Members/metaware/ReferenceDocuments/caspar-description-of-work/at\\_download/file](http://www.casparpreserves.eu/Members/metaware/ReferenceDocuments/caspar-description-of-work/at_download/file))
- [A2] ID1301: Information Model – Interfaces ([http://www.casparpreserves.eu/other-caspar-products/other-caspar-products/caspar-id1301-mod-0001-0\\_1.pdf](http://www.casparpreserves.eu/other-caspar-products/other-caspar-products/caspar-id1301-mod-0001-0_1.pdf) )
- [A3] CASPAR D2201 Preservation DataStore Interface  
([http://www.casparpreserves.eu/Members/cclrc/Deliverables/preservation-datastore-interface/at\\_download/file](http://www.casparpreserves.eu/Members/cclrc/Deliverables/preservation-datastore-interface/at_download/file) )
- [A4] CASPAR D2301 Report on OAIS Access Model  
([http://www.casparpreserves.eu/Members/cclrc/Deliverables/report-on-oais-access-model/at\\_download/file](http://www.casparpreserves.eu/Members/cclrc/Deliverables/report-on-oais-access-model/at_download/file) )
- [A5] CASPAR D4102 Integrated report of R&D activities on the Cultural, Performing Arts and Science data Preservation Activities  
([http://www.casparpreserves.eu/Members/cclrc/Deliverables/integrated-report-of-r-d-activities-on-the-cultural-performing-arts-and-science-data-preservation-activities/at\\_download/file](http://www.casparpreserves.eu/Members/cclrc/Deliverables/integrated-report-of-r-d-activities-on-the-cultural-performing-arts-and-science-data-preservation-activities/at_download/file) )

#### Reference documents

- [R1] CASPAR proposal, Sept 2005
- [R2] OAIS Reference Model (<http://public.ccsds.org/publications/archive/650x0b1.pdf> )

### 1.4 GLOSSARY

|        |  |
|--------|--|
| [Ax]   | Applicable Document  |
| [Rx]   | Reference Document   |
| CASPAR | Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval |
| DoW    | Description of Work  |
| EC     | European Commission  |
| EPM    | Executive Project Management   |
| IPC    | IP Coordinator   |
| IST    | Information Society Technologies   |
| PACP   | Partner Administrative Contact Point   |
| PO     | Project Officer  |
| PPR    | Project Progress Report  |
| PQE    | Project Quality Engineer   |
| PTCP   | Partner Technical Contact Point  |
| R&D    | Research and Development   |
| SQE    | Stream Quality Engineer  |
| ST     | Stream   |
| TN     | Technical Note   |
| WP     | Work Package   |





|                                    |  |
|------------------------------------|--|
| WPL                                | Work Package Leaders   |
| Designated Community               | An identified group of potential Consumers who should be able to understand a particular set of information. The Designated Community may be composed of multiple user communities. (OAIS definition)  |
| 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. (OAIS definition)  |
| Content Information                | The set of information that is the original target of preservation. It is an Information Object comprised of its Content Data Object and its Representation Information. An example of Content Information could be a single table of numbers representing, and understandable as, temperatures, but excluding the documentation that would explain its history and origin, how it relates to other observations, etc. (OAIS definition) |
| Knowledge Base                     | A set of information, incorporated by a person or system, that allows that person or system to understand received information. (OAIS definition)  |
| Representation Information         | The information that maps a Data Object into more meaningful concepts. An example is the ASCII definition that describes how a sequence of bits (i.e., a Data Object) is mapped into a symbol. (OAIS definition)   |





## 2 OAIS MODELS

OAIS [R2] touches on many aspects of relevance to digital preservation but two models are dealt with extensively, namely the Information Model and the Functional Model. The OAIS Infrastructure which is the topic of this document is organised around these two models. The Data Flow models are not dealt with here.

Of course the OAIS concepts have been carefully defined to cover all types of information – which is itself defined very generally as “Any type of knowledge that can be exchanged. In an exchange, it is represented by data”. This domain independence is very important, and distinguishes OAIS from other standards related to digital preservation.

The infrastructure we wish to define and implement must, in order to be very widely usable, be independent of the type of the information being preserved; use of the OAIS concepts help considerably in this aim. A simple analogy would be that a file storage system does not care what the files contain – the storage service is “domain independent”. However, as noted in the CASPAR Conceptual Model [D1201], there must of course be dependence on the information instances – we will refer to this as “domain dependence” in future.

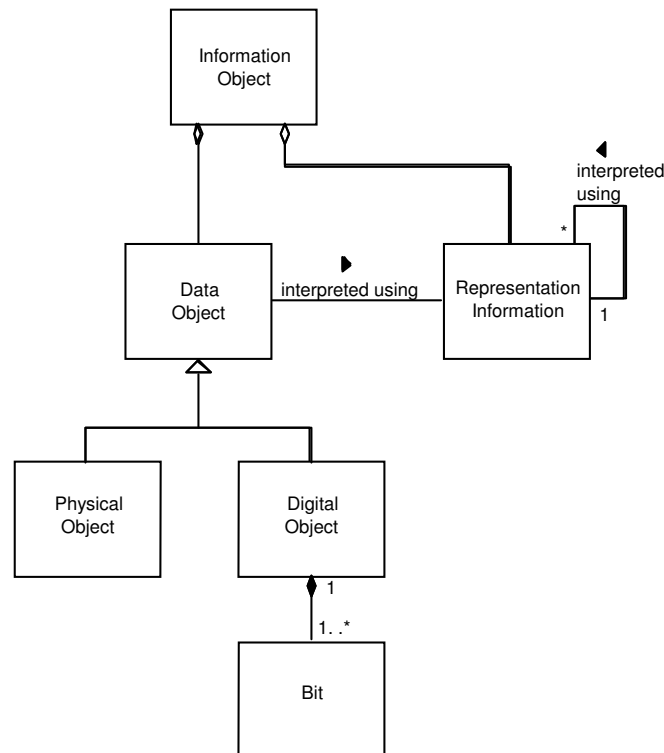
The approach taken in CASPAR is to try to isolate this dependence as far as possible, and in particular we attempt to isolate this domain dependence to the creation of Representation Information and the creation of Provenance which are isolated in collections of tools (Toolkits).





### 3 OAIS INFORMATION MODEL

The OAIS Information Model provides the concepts to support the long-term understandability of the preserved data. This introduces the idea of Representation Information.



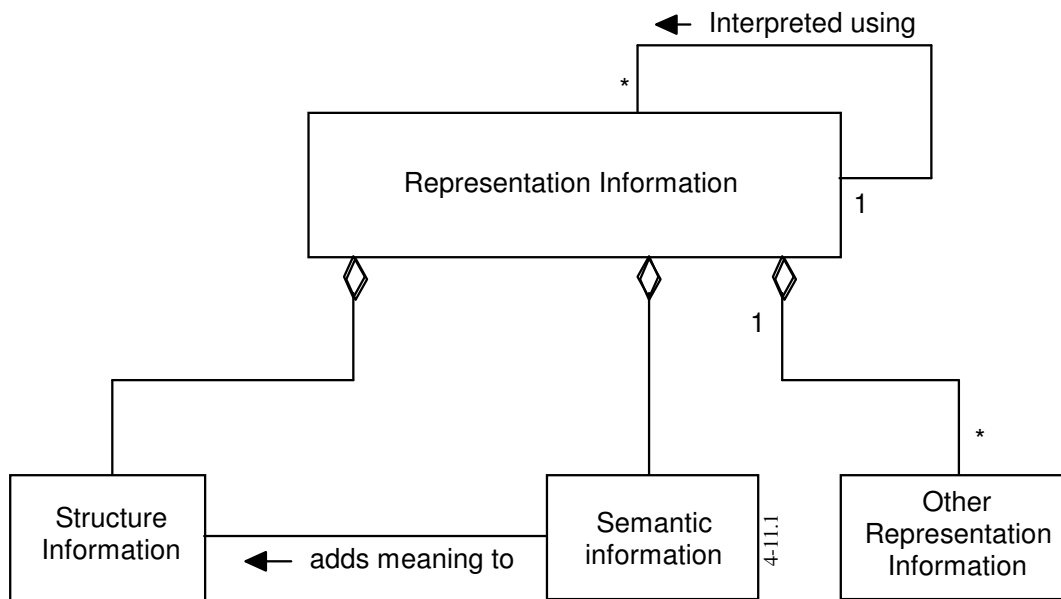
**Figure 1 OAIS Information Model**

The UML diagram in Figure 1 means that

- an Information Object is made up of a Data Object and Representation Information
- A Data Object can be either a Physical Object or a Digital Object. An example of the former is a piece of paper or a rock sample.
- A Digital Object is made up of one or more Bits.
- A Data Object is interpreted using Representation Information
- Representation Information is itself interpreted using further Representation Information

This figure shows that Representation Information may contain references to other Representation Information. When this is coupled with the fact that Representation Information is an Information Object that may have its own Digital Object and other Representation Information associated with understanding each Digital Object, as shown in a compact form by the “*interpreted using*” association, the resulting set of objects can be referred to as a Representation Network.



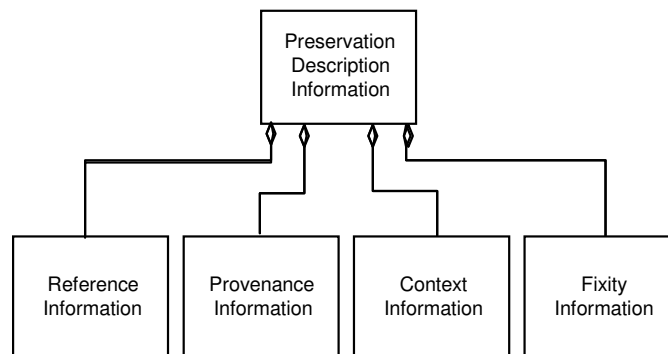


**Figure 2 Representation Information Object**

Figure 2 shows more details and in particular breaks out the semantic and structural information as well as recognising that there may be “Other” representation information such as software.

The types of Representation Information are very diverse and it is highly likely to be discipline dependent, although there will be some commonalities.

### 3.1 PRESERVATION DESCRIPTION INFORMATION



**Figure 3 Types of Preservation Description Information**

Preservation Description Information, including Fixity, Reference, Context and Provenance, will be detailed below. Many aspects are very likely to be discipline independent, for example Fixity, Reference and some aspects of Provenance. However it is likely that at least some aspects of Provenance will be discipline dependent, as will be Context Information.

Work has been done to identify the sources of this type of PDI as reported on in [A5], from which Figure 4 is taken.

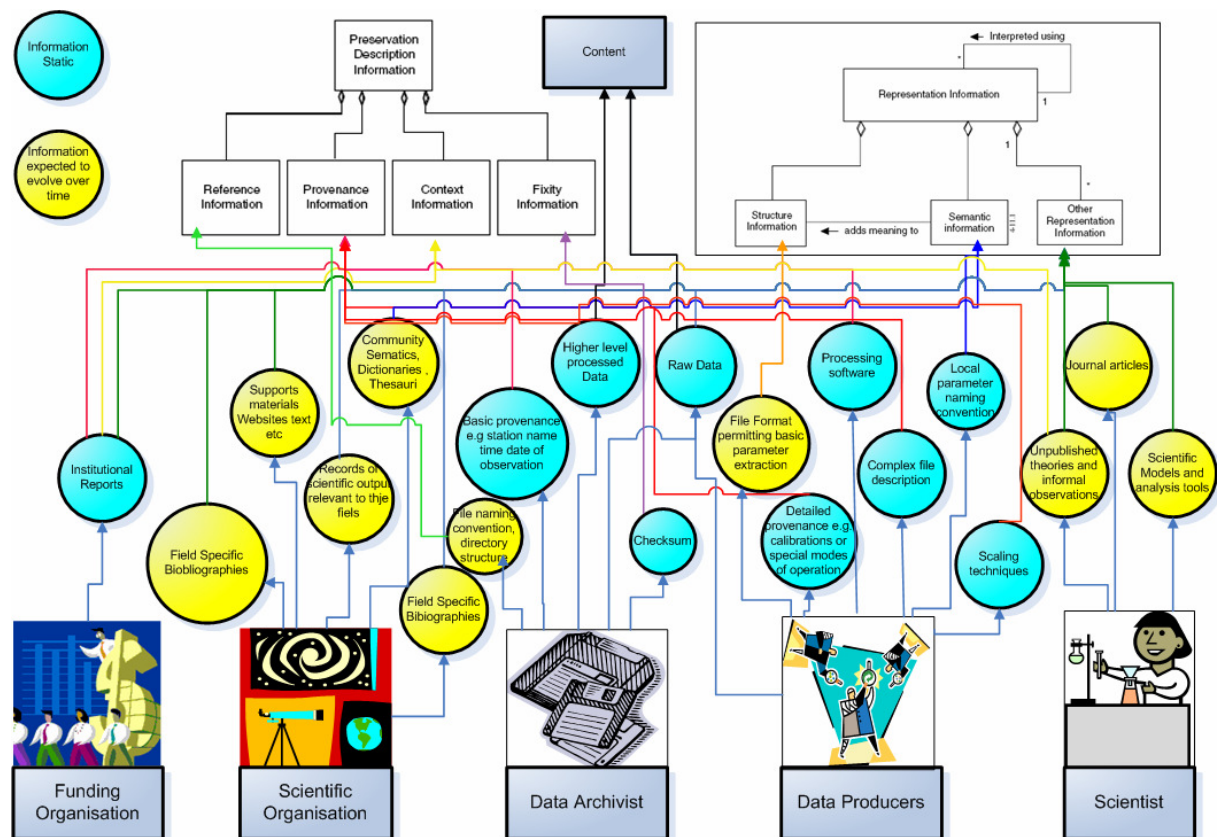


Figure 4 Example of sources of PDI and RepInfo

### 3.1.1 Fixity Information

OAIS defined Fixity Information as the:

*information which documents the authentication mechanisms and provides authentication keys to ensure that the Content Information object has not been altered in an undocumented manner. An example is a Cyclical Redundancy Check (CRC) code for a file.*

*This information provides the Data Integrity checks or Validation/Verification keys used to ensure that the particular Content Information object has not been altered in an undocumented manner. Fixity Information includes special encoding and error detection schemes that are specific to instances of Content Objects. Fixity Information does not include the integrity preserving mechanisms provided by the OAIS underlying services, error protection supplied by the media and device drivers used by Archival Storage. The Fixity Information may specify minimum quality of service requirements for these mechanisms*

Fixity is relevant within the repository or in the transfer phase, but cannot be itself the guarantee for long-term integrity, because of the problem of obsolescence. There are a large number of object digest/hash/checksum algorithms, such as CRC-32, MD5, RIPEMD-160, SHA and HAVAL, some of which are, at the moment, secure in the sense that it is almost impossible for changes in the digital object to fail to be detected – at least as long as the original digest itself is kept secure. However in the future processing power, of individual processors and of collections of processors, will increase and algorithms may become “crackable”. Warning of the vulnerability of any particular type of digest algorithm would be a function of the Orchestration manager.



In a broad sense the tools for fixity used by the repositories (and by the creator) have to be documented and this documentation (specifically related to the process and to the responsibilities) will be part of the PDI component and would play a relevant role for ensuring the trustworthiness (integrity as a part of it) of the preserved resources.

The CASPAR Key Store concept – which is a Registry/Repository (see below) – could provide additional security for the digests. It may be possible to use one object digest as an identifier to be sent to the Key Store which returns the other digest which can be used to confirm the fixity of the object. This functionality has not yet been implemented.

### 3.1.2 Reference Information

OAIS defines Reference Information as the information which:

*identifies, and if necessary describes, one or more mechanisms used to provide assigned identifiers for the Content Information. It also provides those identifiers that allow outside systems to refer, unambiguously, to this particular Content Information. Examples of these systems include taxonomic systems, reference systems and registration systems. In the OAIS Reference Model most if not all of this information is replicated in Package Descriptions, which enable Consumers to access Content Information of interest.*

The identifiers must be persistent and are referred to here as Persistent Identifiers, and are unique in that an identifier should be usable to locate the specific digital object with which it is associated, or an identical copy of that object. Curation Persistent Identifiers (CPID) are reported on in [A4], with some additional remarks on the interface in section 3.4.1.1 below.

### 3.1.3 Context Information

*This information documents the relationships of the Content Information to its environment. This includes why the Content Information was created and how it relates to other Content Information objects existing elsewhere.*

Context covers an extremely broad range of topics and it is difficult to define a precise boundary. In fact Provenance Information, described next, can be viewed as a special type of Context Information.

We do not expect to formalise Context information because of its great variety. However each piece of Context information should have its own Representation Information.

### 3.1.4 Provenance Information

*This information documents the history of the Content Information. This tells the origin or source of the Content Information, any changes that may have taken place since it was originated, and who has had custody of it since it was originated. This gives future users some assurance as to the likely reliability of the Content Information.*

A significant consideration is the inheritance of Provenance, in that given a digital object with a certain Provenance there are a number of directly related objects which share the Provenance of that object, including:

- A copy of the object – which will have identical Provenance plus an additional event, namely the copy process which created it
- An object derived from the original object – plus perhaps several others. In this case the Provenance of the new object inherits Provenance from its “parents”, and has a new event, namely the process by which it was created.

An important question which needs to be tackled is the extent to which we could or should avoid duplications of the Provenance entries. It is worth noting that this question comes to the fore with digital, as opposed to physical, objects.





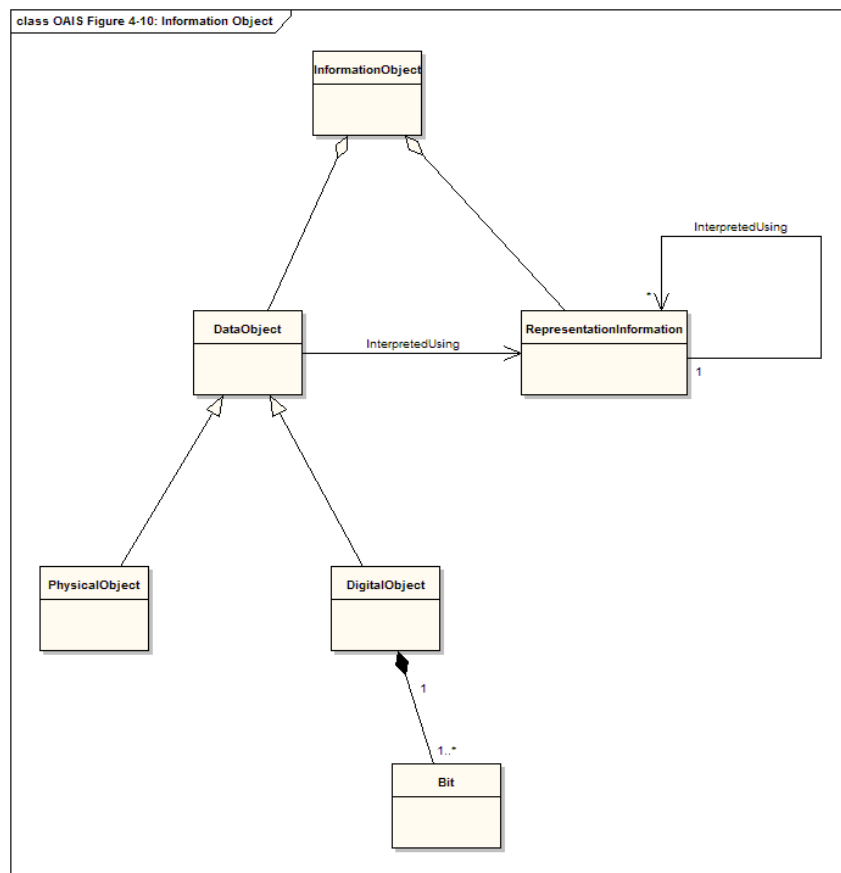
Further information about Provenance is provided in [D2301] where work is beginning on defining an information model and associated interfaces..

### 3.2 EXTENDING THE OAIS INFORMATION MODEL

The Information Model in OAIS is not sufficiently detailed to be implemented in software, however it can be used to generate a *Platform Independent Model* (PIM<sup>1</sup>) and, adding the obvious methods to allow one to *get* and *set* the various sub-components, to create an API to support the fundamental concepts of OAIS. The following sections give an outline of the route from the OAIS document to the API.

### 3.3 OAIS INFORMATION MODEL PIM

The PIM can be shown as follows:



**Figure 5 PIM of OAIS Information Model**

This is of course essentially identical to the OAIS UML diagram, and needs no further discussion.

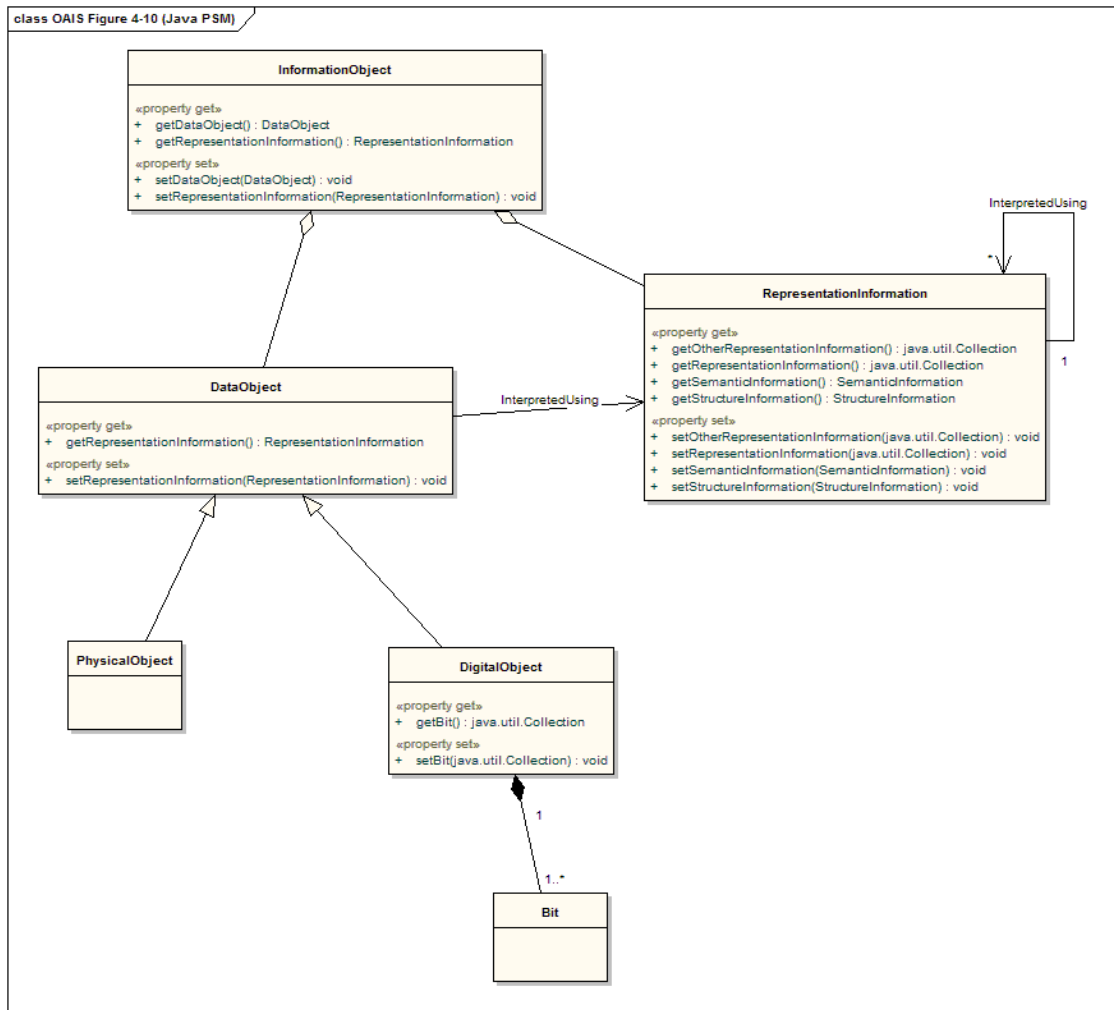
From the PIM the *Platform-Specific Model* (PSM<sup>2</sup>) may be created using automated tools<sup>3</sup>.

<sup>1</sup> A platform-independent model or PIM is a model of a software or business system that is independent of the specific technological platform used to implement it – see [http://en.wikipedia.org/wiki/Platform-independent\\_model](http://en.wikipedia.org/wiki/Platform-independent_model)

<sup>2</sup> A platform-specific model is a model of a software or business system that is linked to a specific technological platform (e.g. a specific programming language, operating system or database) – see [http://en.wikipedia.org/wiki/Platform-specific\\_model](http://en.wikipedia.org/wiki/Platform-specific_model)

<sup>3</sup> CASPAR uses Enterprise Architect – see <http://www.sparxsystems.com.au/>





**Figure 6 PSM (Java) of OAIS Information Model**

The methods such as *getRepresentationInformation()* provide a way to obtain the desired component – in this case it would return the Representation Information which forms part of the Information Object. From this we have the interfaces:

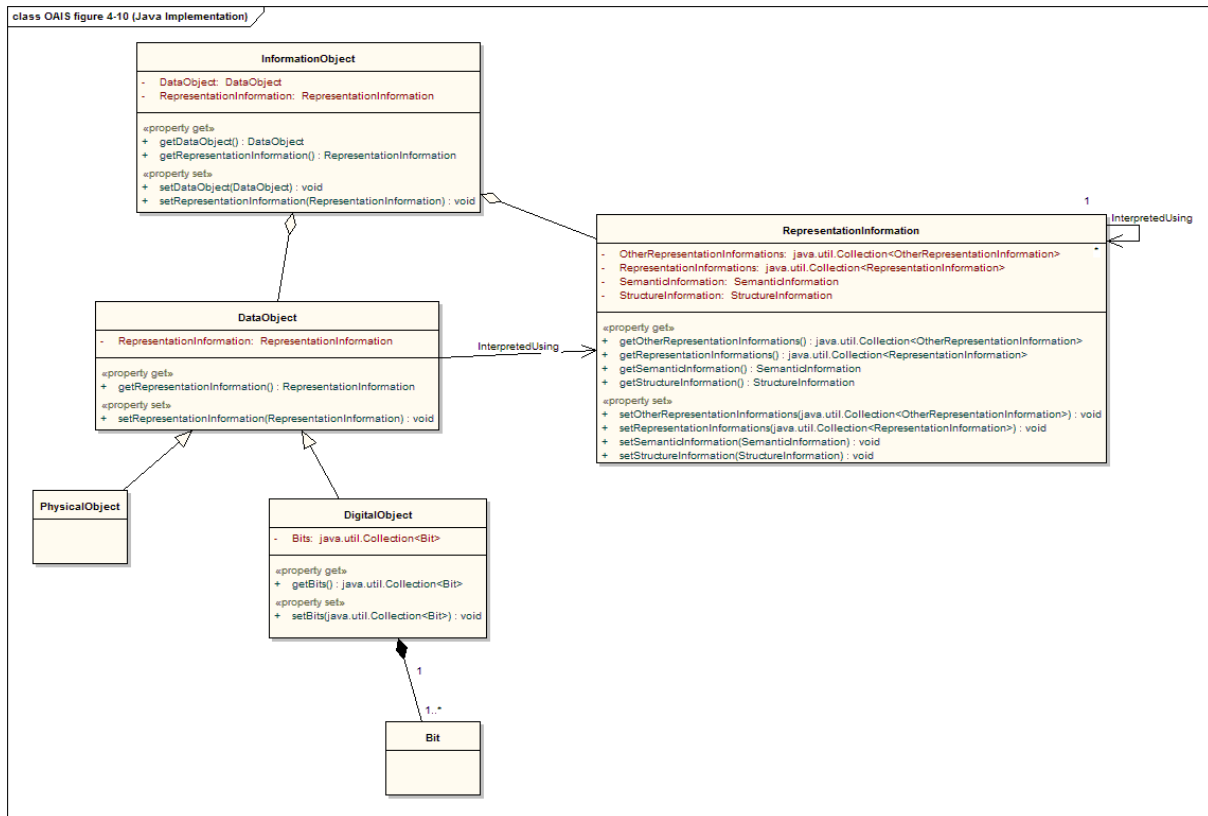


Figure 7 OAIS Information Model - interfaces

### 3.4 OAIS INFORMATION MODEL INTERFACES

The next step involves introducing new concepts which, while we believe them to be eminently reasonable, are nevertheless fairly arbitrary choices.

The following are extracted from ID1301:INFORMATION MODEL – INTERFACES [A2], where further details are available.

#### Operations

| Method   | Notes | Parameters                                       |
|--|-------|--|
| <b>getDataObject()</b> DataObject<br>Public                                  |       |  |
| <b>getRepresentationInformation()</b><br>RepresentationInformation<br>Public |       |  |
| <b>setDataObject()</b> void<br>Public  |       | <b>DataObject</b> [in]_dataObject                |
| <b>setRepresentationInformation()</b> void<br>Public                         |       | <b>RepresentationInformation</b> [in]<br>repInfo |



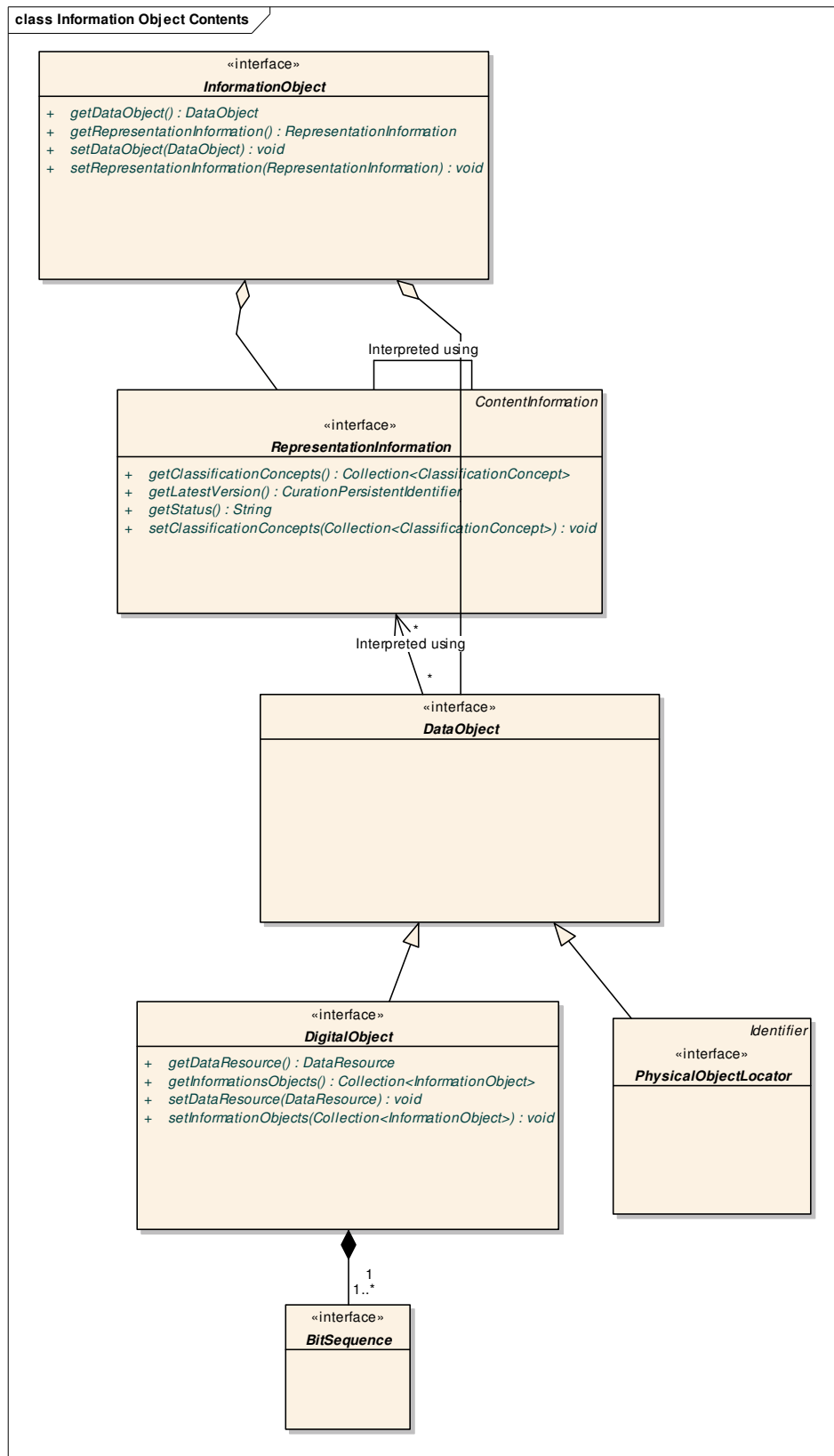


Figure 8 Information Object Interfaces

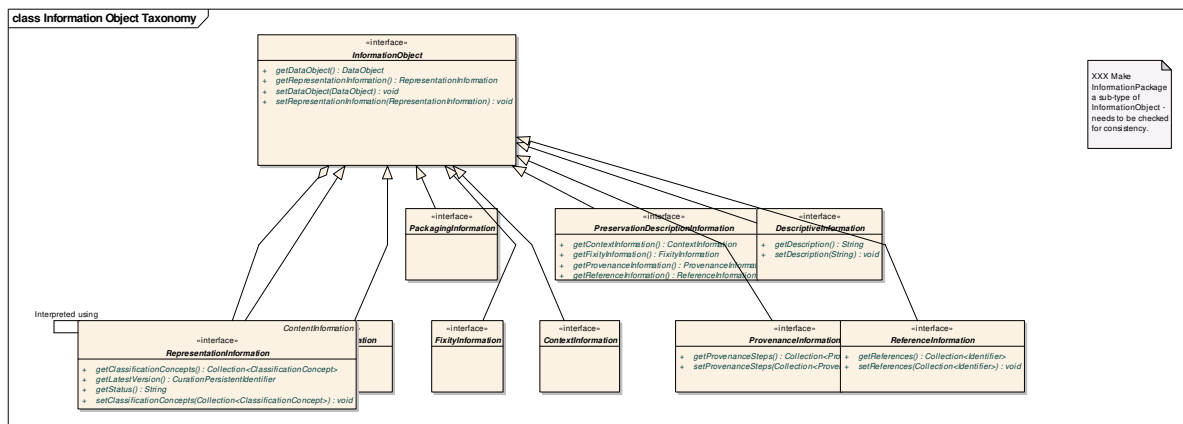


Figure 9 Information Object Taxonomy

### 3.4.1 Additional concepts

Examples of additional concepts which we introduce, and for which we attempt to define rather general interfaces and specialisations, include Identifier and Messaging. The following sections discuss the reasoning behind these ideas.

#### 3.4.1.1 Identifier

An Identifier allows us to name, and in particular locate, something – either a digital object or a physical object. It would be tempting to adopt a single existing identifier system, such as DOI, or to invent yet another identifier system. However the very plethora of such systems does demonstrate that adopting or inventing a single identifier system will almost certainly NOT stand the test of time. One may dominate in a particular discipline or for a particular time, but if we want something to be usable for all types of digitally encoded information and for the long term then we cannot rely on a single system.

If in fact, by some miracle, a single system is universally adopted and which lasts over time, then all we have done is to introduce a little additional complexity, and have not prevented the use of that system.

The common features which we need from an identifier is to provide us with one or more *Locators*, by which we mean something which gives us an object – let us say a *String* of characters, so it can be rendered as something which is human readable, encoded in some digital encoding, such as Unicode. In order to know what to do with this String, for example DOI:10.123456, we need some clue. This clue we refer to as a *Resolver*, which again we assume is a String.

Thus we are led to the basic interface to the Identifier – *getLocators*. This returns a collection of Locators. Each Locator has a *getValue* and *getResolver* method.

In addition we need some way to tell whether one identifier is a later version than another, which in a JAVA implementation may be covered by implementing the Comparable interface<sup>4</sup>.

<sup>4</sup> see for example <http://java.sun.com/j2se/1.4.2/docs/api/java/lang/Comparable.html>





This Identifier is rather simple, and we believe that it is useful to introduce some specialisations, namely

- Persistent Identifier – an identifier which one hopes will be usable over the long term to identify digital objects i.e. objects which may be exactly copied. This means that locators may resolve to completely different physical locations, all of which have identical copies.
- Physical Object Identifier – an identifier which applies to a physical object – which are different from digital objects in that the latter can be copied exactly whereas physical objects cannot be copied and hence for which a unique physical location must be specified. Note that many different character strings may point to the same location e.g. a position of the surface of the earth may be specified by a Lat/Long pair or Street address or Map reference..
- Info Object Identifier – an identifier for information objects which may not be persistent, and may for example provide relative, e.g. within the same file or file system, rather than absolute locations.

Other specialisation may be useful in future.

We also define a *Curation Persistent Identifier* as a specialisation of Persistent Identifier which applies to Representation Information; further details are provided in section 3.6.

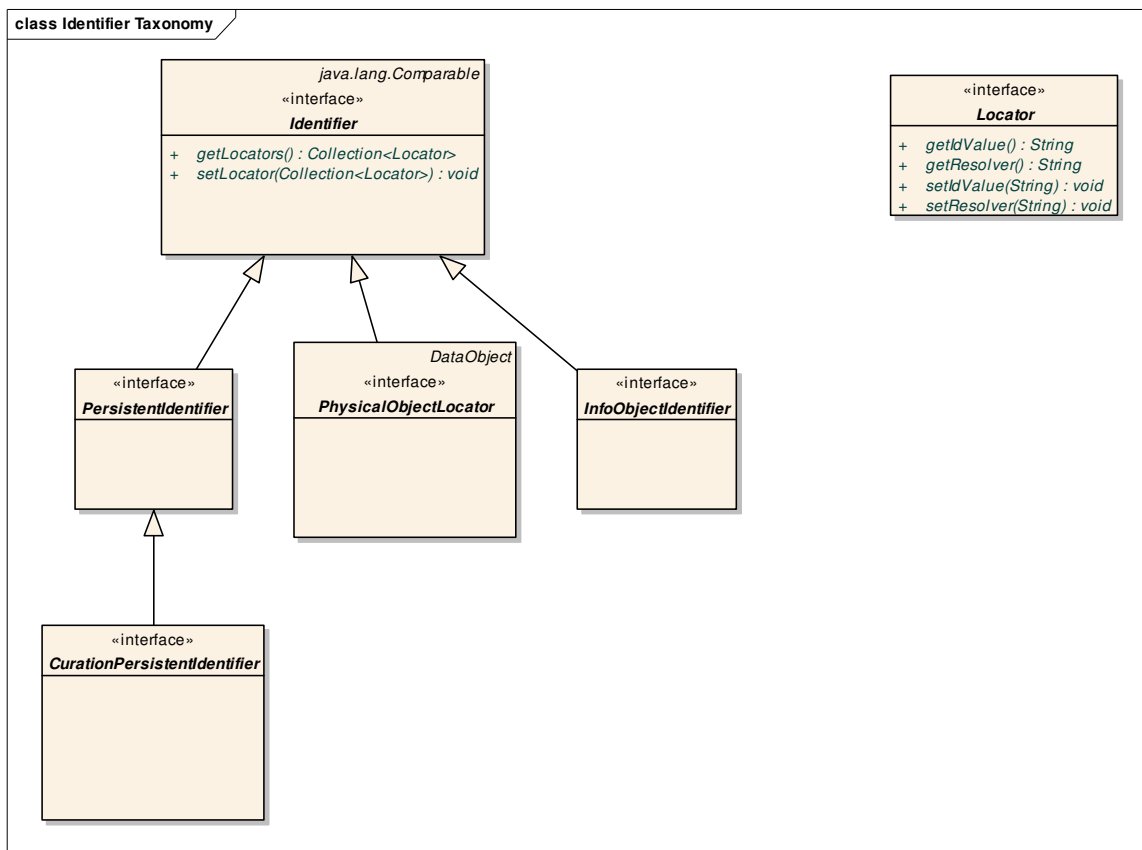


Figure 10 Identifier Taxonomy

Operations

| Method                               | Notes | Parameters |
|--------------------------------------|-------|------------|
| getLocators()<br>Collection<Locator> |       |            |



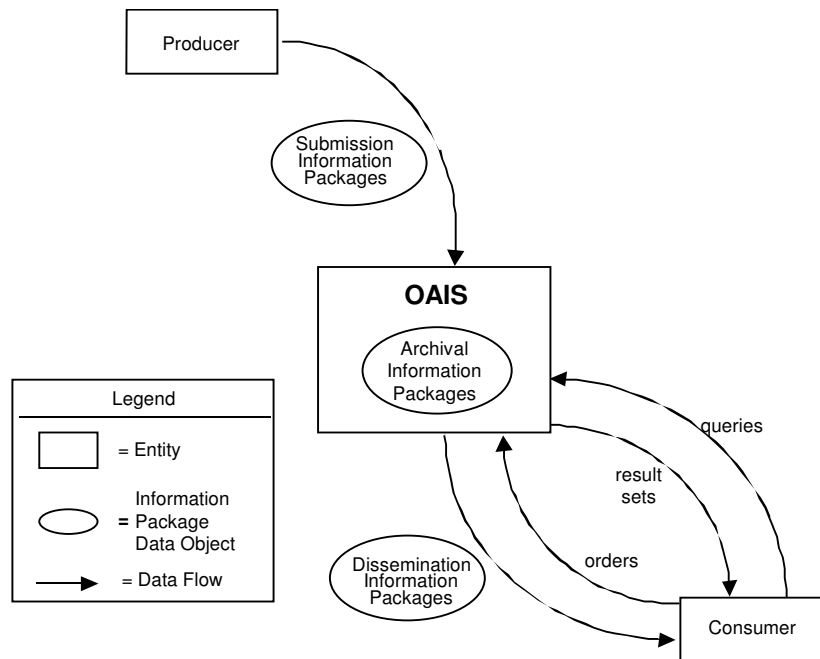


| Method                             | Notes | Parameters                                     |
|------------------------------------|-------|--|
| Public                             |       |  |
| <b>setLocator()</b> void<br>Public |       | <b>Collection&lt;Locator&gt;</b> [in]_locators |

### 3.4.2 Messaging

The messaging referred to here is not the low level messaging protocols such as might be associated with Web Services, the details of which are (almost certainly) transient; instead we mean the information passed from one entity to another, some instances of which may be persistent.

In more general terms a message is a piece of information which is **produced** and sent to someone or something. If the message is treated as ephemeral by the receiver then there is no further issue from the point of view of preservation. On the other hand if the receiver wishes to preserve the message then the receiver must act as an OAIS. Thus if we wish to deal with Persistent Messages we can use Figure 11 which is reproduced from the OAIS Reference Model.



**Figure 11 OAIS External Data Flows**

With this in mind, the approach we take is to treat messages as Submission Information Packages (SIPs), which are sent from a Producer to an “Archive” as shown in Figure 11.

The SIP is defined in very inclusive, broad, terms in OAIS. The most important capability is that an AIP may be constructed from one or more of these SIPs and the AIP contains everything needed or long term preservation, fulfilling an important aspect of persistence.

There are various ways of providing this capability. The most obvious way is demanding that each message is itself a full AIP, however this would impose rather heavy demands on the underlying messaging system, and of course many messages are transient. The way we have chosen avoids this by instead demanding that each message has a unique identifier (an *InfoObjectIdentifier*) which can be





used to ask the sender for the additional information needed in order to make the message (a specialisation of SIP) an AIP.

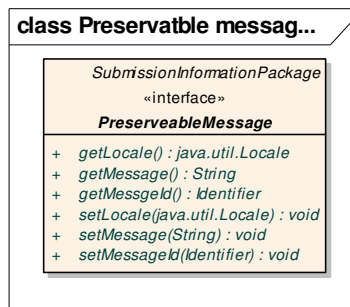


Figure 12 Preservable Message

| Method   | Notes   | Parameters                       |
|--|---|----------------------------------|
| <b>getLocale()</b><br>java.util.Locale<br>Public | Locale for the text in Message  |                                  |
| <b>getMessage()</b> String<br>Public             | The simplest type of message has ContentInformation which is a text message.                              |                                  |
| <b>getMessageId()</b> Identifier<br>Public       | The message may have an associated identifier which may be used to obtain the message in other languages. |                                  |
| <b>setLocale()</b> void<br>Public                | Set the Locale for the text Message.  | <u>java.util.Locale</u> [in]_loc |
| <b>setMessage()</b> void<br>Public               | Simple text message   | <u>String</u> [in]_msg           |
| <b>setMessageId()</b> void<br>Public             | Set the identifier which may point to the message in alternative languages.                               | <u>Identifier</u> [in]_msgid     |

Thus if the recipient of the message wishes to preserve that message, it can use the MessageId to request further information from the sender, such as Representation Information and PDI (e.g. Provenance and Context).

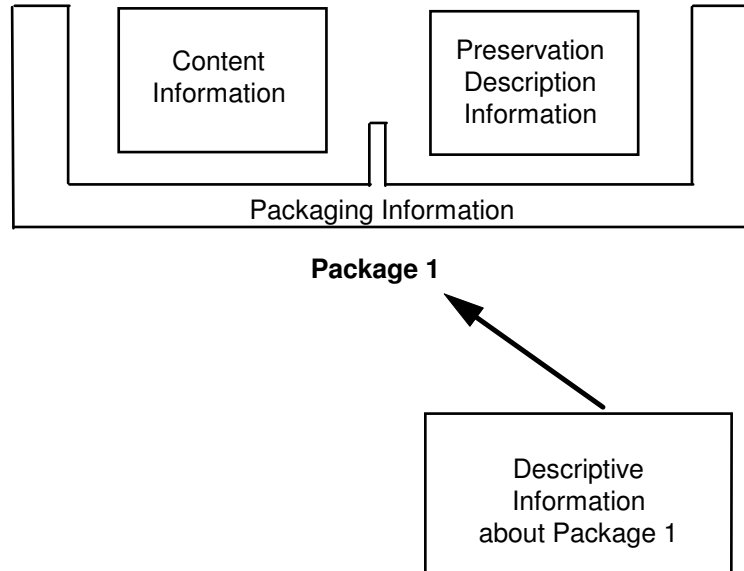
### 3.5 PACKAGING

OASIS Packaging Information is that information which

*either actually or logically, binds or relates the components of the package into an identifiable entity on specific media. For example, if the Content Information and PDI are identified as being the content of specific files on a CD-ROM, then the Packaging Information may include the ISO*

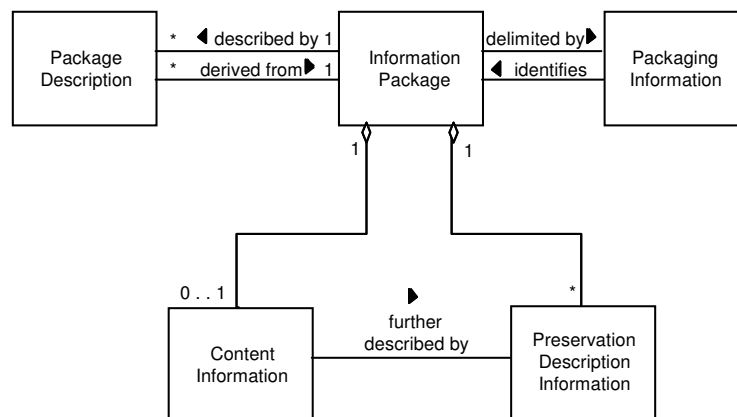


9660 volume/file structure on the CD-ROM. These choices are the subject of local archive definitions or conventions. The Packaging Information does not necessarily need to be preserved by an OAIS since it does not contribute to the Content Information or the PDI. However, there are cases where the OAIS may be required to reproduce the original submission exactly. In this case the Content Information is defined to include all the bits submitted.



**Figure 13 Packaging concepts**

The contents of a general Information Package is illustrated in Figure 14



**Figure 14 Information Package contents**

OAIS further introduced a taxonomy of Information Packages, as shown in Figure 15





### 3.5.2 API for packages

Following a similar methodology as for the Information Object, we have produced a set of interfaces for packages which allow one to obtain the various sub-components, in a fairly obvious way. In addition to these we have added *getVersion()* which allows one to find the version of the package – in addition to also implementing the *java.lang.Comparable* interface, as for Identifier.

At the moment the only additional method which the AIP provides is the boolean *isValid()* which returns *TRUE* if and only if the AIP is valid i.e. has all the mandatory components; note that the DIP and SIP do not have any mandatory components.

## INFORMATIONPACKAGE

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.

### Methods

| Method   | Notes  | Parameters   |
|--|--|--|
| <b>getContentInformation()</b><br>ContentInformation<br>Public     |  |  |
| <b>getPackageDescription()</b><br>PackageDescription<br>Public     |  |  |
| <b>getPackagingInformation()</b><br>PackagingInformation<br>Public |  |  |
| <b>getPDI()</b><br>PreservationDescriptionInformation<br>Public    |  |  |
| <b>getVersion()</b> Version<br>Public                              | An opaque object which may be used by the Comparable interface |  |
| <b>setContentInformation()</b> void<br>Public                      |  | <u>ContentInformation</u> [in]<br>contentInformation |
| <b>setPackageDescription()</b> void<br>Public                      |  | <u>PackageDescription</u> [in]<br>packageDescription |
| <b>setPackagingInformation()</b> void                              |  | <u>PackagingInformation</u> [in]                     |





|               |  |   |
|---------------|--|---|
| Public        |  | packagingInformation                      |
| setPDI() void |  | <b>PreservationDescriptionInformation</b> |
| Public        |  | [in]_pdi                                  |

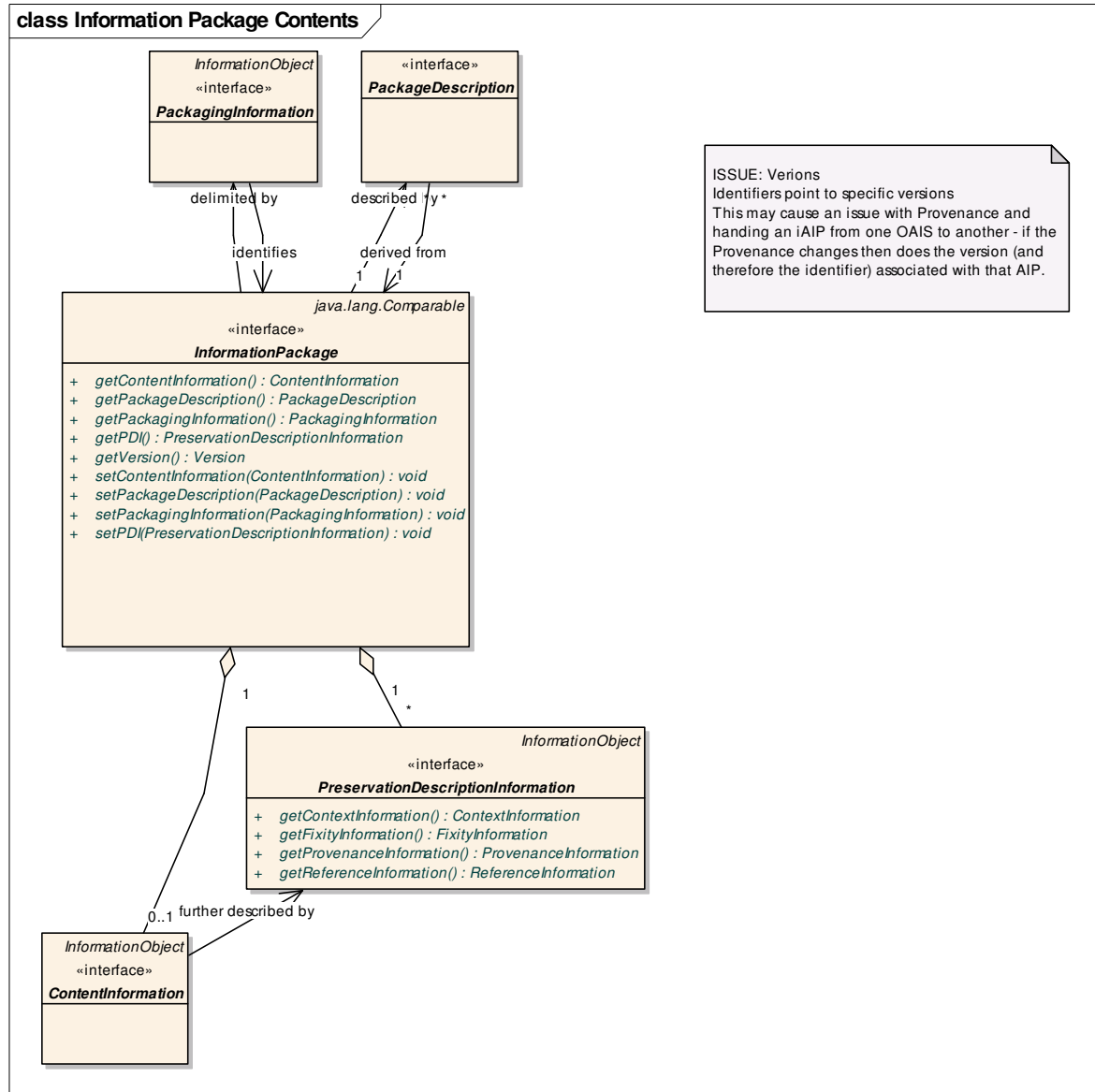


Figure 17 Information Package Contents







### 3.6 REPRESENTATION INFORMATION

Representation Information is a key concept in OAIS and tools must be available to create it. However the variety of tools is such that it has not proved possible to define a general API other than a very simple one along the lines of *createRepInfo()*.

Examples of the many types of data and their Representation Information is provided in [A5].

#### 3.6.1 RepInfo toolbox

The Representation Information toolbox is simply a GUI to simplify access to a collection of such tools.

The tools in the collection would be expected to grow over time and since Representation Information may consist of just about anything - definitions of virtual machines, software, physical documents, descriptions of data (EAST, XML, DRB etc) – the corresponding tools can be equally varied.

Given this variety, the toolbox cannot be prescriptive but can provide suggestions about the type of tools which may be useful in creating the appropriate Representation Information.

### 3.7 REGISTRY/REPOSITORY OF REPRESENTATION INFORMATION

While OAIS does not refer explicitly to Registries of Representation Information, the focus on the Representation Network does allow one to make a reasonable argument in their favour as follows.

- At any point into the future there must be adequate Representation Information in the AIP for the Designated Community to be able to understand and use the Data Object.
- As the KnowledgeBase of the Designated Community changes over time, the amount of Representation Information which must be available must therefore grow over time
- No organisation has unlimited resources with which to collect the required Representation Information, therefore they must have access to such information from other sources
- A central store would facilitate access and sharing of Representation Information.

Such a store must be an OAIS repository and will be referred to as a Registry/Repository to emphasise the fact that it contains, i.e. is a Repository for, Representation Information Objects, some of which may be very large. The basic functionality is:

- to be able to store Representation Information, and provide an identifier to be able to retrieve it
- to be able to supply the appropriate piece of Representation Information given such an identifier
- to be able to respond to queries about the Representation Information it holds.

#### 3.7.1 Networks of Representation Information

The concept of a Representation (Information) Network (OAIS section 4.2.1.3.2) is an important one in long term maintenance of understandability, and so CASPAR must implement these.

In order to implement such a network we attach to each and every piece of Representation Information further Representation Information – or rather a pointer to further Representation Information. In principle each piece of Representation Information is an opaque object (either physical or digital) and one proceeds through the network until one finds something that one understands and then one reverses the process until one understands the original piece of Representation Information.

In order to simplify the process we have created an artefact as an initial step. This is itself a piece of Representation Information with its own Representation Information, and hence is consistent with everything else.

By making the initial piece of Representation Information a simple XML file we have something which, for a little while at least, is readable by a fairly wide audience. We will refer to this as a *RepInfoLabel*. An example is as follows.





```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rilabel xmlns=http://registry.dcc.ac.uk/dcc-rilabel2.xsd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://localhost/dcc-rilabel.xsdhttp://registry.dcc.ac.uk/dcc-rilabel2.xsd">
<description>MST known file type</description>
<timestamp>Mon May 12 23:09:56 BST 2008 </timestamp>
<representationinformation>
  <semantic>
    <cpid>
      <description>Semantic Information about MST version 3 NetCDF
data files </description>
      <value>urn:uuid:40e0c3de-a405-4759-b116-eda15d77df59</value>
    </cpid>
  </semantic>
  <structure>
    <cpid>
      <description>RepInfo Structural description of MST NetCDF data
in DRB XML schema format</description>
      <value>urn:uuid:61c0ef4d-e44e-49bb-842d-cf856d8b4a36</value>
    </cpid>
  </structure>
  <other/>
</representationinformation>
</rilabel>

```

For simplicity we have omitted the *Resolver* details for the CPID.

The CPID under *<structure>*, for example, points to an object which provides details of the format of the data file to which this RepInfoLabel is attached. This description is, say, a DRB description of a NetCDF<sup>5</sup> file. This description itself has a CPID attached which provides details of the DRB description language (structure plus semantics).

### 3.7.2 Knowledge Management

A key type of Representation Information is Semantic Information; this is neglected in most other digital preservation projects. Some of the work we have undertaken is described in [A5]. Of particular importance is the link to the Knowledge Base of the Designated Community, a fundamental concept from OAIIS. A key question is whether or not it is possible to define such a Knowledge Base; if this is not possible then the OAIIS concepts must be revised. Such a Knowledge Base also defines the amount of Representation Information that is needed in the Archival Information Package.

In addition Knowledge management techniques play a fundamental role in CASPAR which is closely tied to the overall issue of Knowledge Management, which is more described very fully in [D2101B] and is touched upon in the next section. The key functionality is to track the dependencies between pieces of Representation Information; based on the information provided by the Registry/Repository about the Representation Network that it holds.

<sup>5</sup> <http://www.unidata.ucar.edu/software/netcdf/>



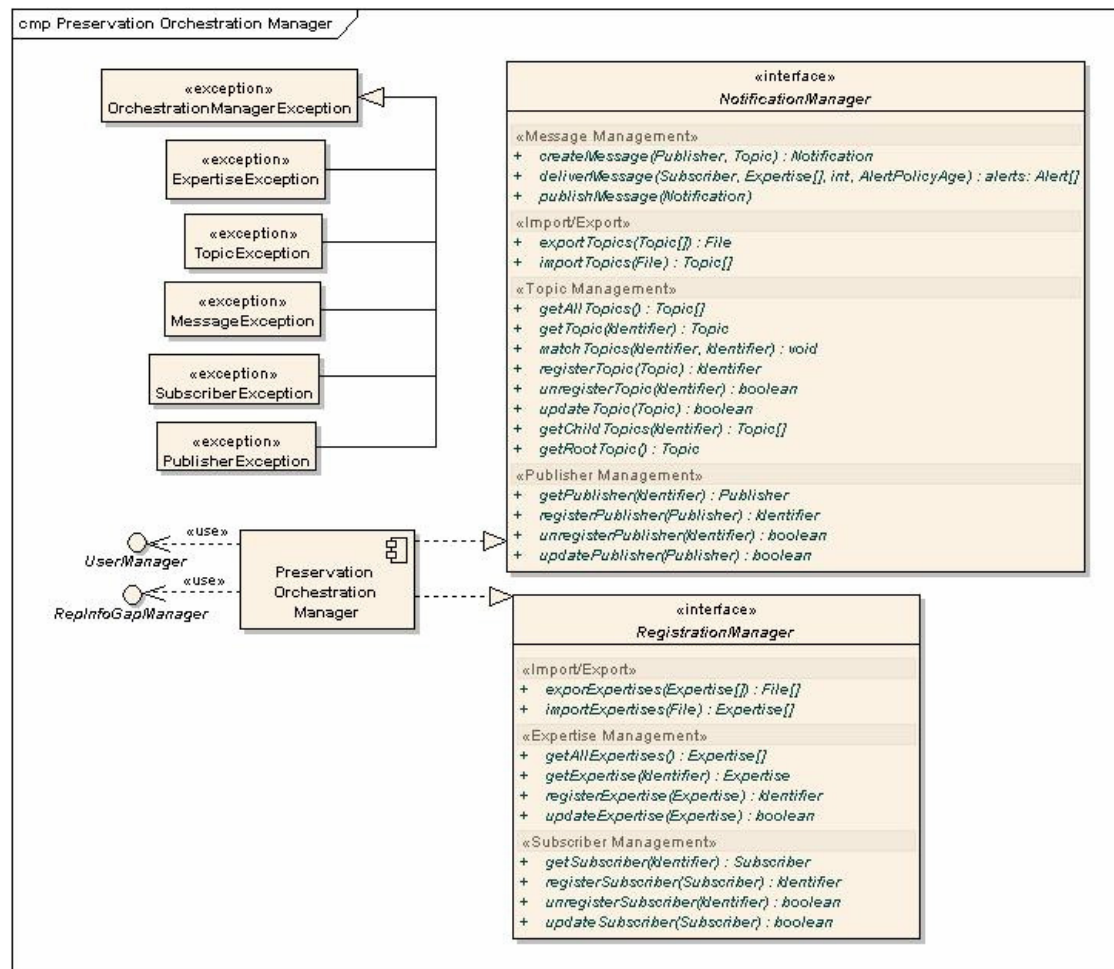
### 3.7.3 Obtaining additional Representation Information

The Orchestration Manager which CASPAR introduces has no counterpart in OAIS. It is introduced to facilitate the sharing of information.

The basic functionality is to

- allow users to register with the service, identify themselves and register their expertise
- deal with messages, both from users about specific topics and to registered users who have the appropriate expertise to help develop, for example, the appropriate Representation Information.

An internal report on the Orchestration Manager is available<sup>6</sup>.



**Figure 20 Orchestration Manager Interfaces**

As the Knowledge Base of the Designated Community changes over time, as it almost certainly will, then “gaps” will arise between the existing Representation Information and the Knowledge Base (for that particular Designated Community). These gaps are identified based on information supplied by the preservation community to the Orchestration Manager about changes that may affect a particular Knowledge Base. In order to “plug the gaps” identified by the CASPAR Gap Manager the Orchestration Manager requests people who have claimed to have the appropriate expertise to create, or locate, the required Representation Information.

<sup>6</sup><http://developers.casparpreserves.eu:8080/hudson/job/CASPAR-POM/ws/implementation/orchestration/etc/html/res/POM-Spec-Ref-1.4.pdf>

## 4 FUNCTIONAL MODEL

The OAIS Functional Model provides a simple model for the functional entities into which an archive may be decomposed. This section looks at each of the functional entities and maps them to CASPAR components.

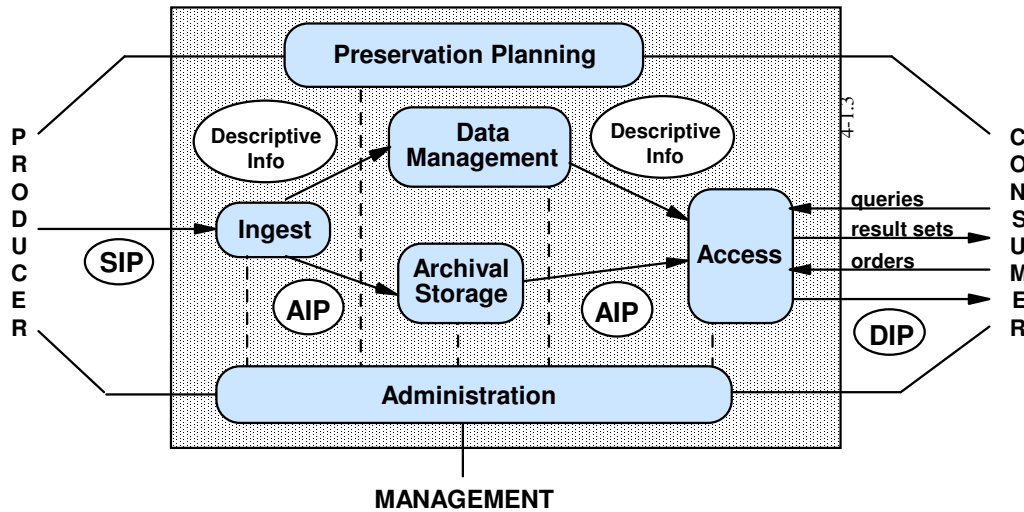


Figure 21 OAIS Functional Model

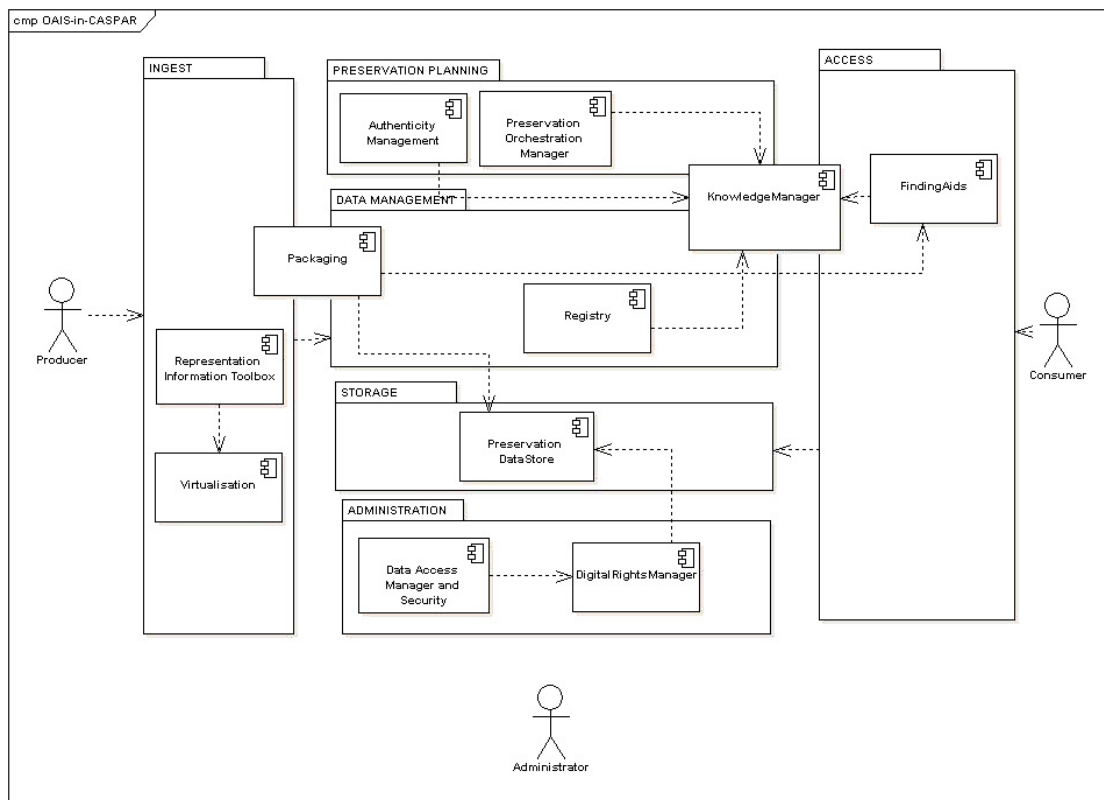


Figure 22 Mapping OAIS Functional Model to CASPAR Key Components



## 4.1 PRESERVATION PLANNING

The Preservation Planning functional entity has a number of responsibilities including *providing the services and functions for monitoring the environment of the OAIIS and providing recommendations to ensure that the information stored in the OAIIS remains accessible to and usable by the Designated User Community over the long term.*

The Registry described above provides part of the capabilities required, including and going beyond simply monitoring the environment. In addition the CASPAR Conceptual Model includes an Orchestration Manager which acts as a (fairly simple) clearing house for information contributed by many other people, thereby sharing the effort of monitoring the many other changes in environment.

## 4.2 DATA MANAGEMENT

*This entity provides the services and functions for populating, maintaining, and accessing both Descriptive Information which identifies and documents archive holdings and administrative data used to manage the archive.*

CASPAR provides no dedicated component for the Data Management function, although it is related to the Packaging and the Knowledge management components. Packaging is involved because the Package Descriptions are used in Data Management to construct its database, and Knowledge Management techniques may be used to interrogate this database.

Nevertheless there is no dedicated component because of the very large variety of functions which archives tend to use their internal databases for. In particular the public face of the archive and the “value-added” services which make archives attractive to users (over and above their preservation capabilities) tend to be driven from this database. Instead of a general component CASPAR instead will have a number of ad-hoc implementations; if may prove possible to derive a more generally useful, tailorable, component after we have gained experience linking to existing archives.

## 4.3 ARCHIVAL STORAGE

*This entity provides the services and functions for the storage, maintenance and retrieval of AIPs. Archival Storage functions include receiving AIPs from Ingest and adding them to permanent storage, managing the storage hierarchy, refreshing the media on which archive holdings are stored, performing routine and special error checking, providing disaster recovery capabilities, and providing AIPs to Access to fulfil orders.*

This function is covered by its own report [A3]. One of the key OAIIS concepts that this implements is that each piece of digitally encoded information held as part of the AIP has its own Representation Information. Another is that the Archival Storage system can automatically take care of recording the provenance about internal activities such as accessing, copying and transforming.

## 4.4 ACCESS

*This entity provides the services and functions that support Consumers in determining the existence, description, location and availability of information stored in the OAIIS, and allowing Consumers to request and receive information products.*

Report [A4] covers the Access component. The DIP is a very loosely defined object in OAIIS. However it is possible to try to define the Representation Information which a DIP should contain in order to be understandable by a particular Designated Community.

## 4.5 INGEST

*This entity provides the services and functions to accept Submission Information Packages (SIPs) from Producers (or from internal elements under Administration control) and prepare the contents for storage and management within the archive.*





As with the Data Management functional entity, there is no dedicated component but instead the Package Manager creates AIPs, which also contain instances of or pointers to, the Representation Information which is created with the RepInfoToolkit.





## 5 CONCLUSIONS

This document summarises the extent to which CASPAR has been able to base implementations on the models in OAIS. The simple UML models have been extended to generate *reasonable* interfaces. While these are still abstract, nevertheless they seem to be usable, and this is being tested by the CASPAR implementations.

Clearly our implementations will not last forever, and perhaps may not last for very long. Yet by striving to tie the work extremely closely to the fundamental concepts of OAIS it is possible to have some hope of longevity of the concepts and interfaces which have been created. It is true that at the time of writing, the project is not fully consistent with these ideas, but it is hoped that this will be achieved in the next iterations.

