Document identifier: CASPAR-D1201-TN-0101-1_0
Submission Date: 15-05-2007
Due date: 15-05-2007
Work package: 1200
Partners: All Partners
WP Lead Partner: CCLRC
Document status: FINAL

Abstract: This document provides the first version of the Conceptual Model which underpins the CASPAR project.
### Document Status Sheet

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Comment</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0_0</td>
<td>13 Mar 2007</td>
<td>Initial draft bringing together contributions</td>
<td>David Giaretta</td>
</tr>
<tr>
<td>0_1</td>
<td>10 Apr 2007</td>
<td>Revision based on comments received including a re-organisation in order to clarify concepts</td>
<td>David Giaretta</td>
</tr>
<tr>
<td>0_2</td>
<td>20 Apr 2007</td>
<td>Revision based on comments received</td>
<td>David Giaretta</td>
</tr>
<tr>
<td>0_3</td>
<td>30 Apr 2007</td>
<td>Revision based on further feedback about treatment Authenticity</td>
<td>David Giaretta</td>
</tr>
<tr>
<td>0_4</td>
<td>5 May 2007</td>
<td>Final draft refining treatment of Virtualisation and namespaces</td>
<td>David Giaretta</td>
</tr>
<tr>
<td>1_0</td>
<td>15 May 2007</td>
<td>Final version</td>
<td>David Giaretta</td>
</tr>
</tbody>
</table>
### Project information

<table>
<thead>
<tr>
<th><strong>Project acronym:</strong></th>
<th>CASPAR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project full title:</strong></td>
<td>Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval</td>
</tr>
<tr>
<td><strong>Proposal/Contract no.:</strong></td>
<td>IST-2006-033572</td>
</tr>
</tbody>
</table>

### Project Officer: Carlos Oliveira

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

### Project Co-ordinator: David Giaretta

| **Address:** | STFC (formerly CCLRC), Rutherford Appleton Laboratory  
Chilton, Didcot, Oxon OX11 0QX, UK |
| **Phone:** | +44 1235 446235 |
| **Fax:** | +44 1235 446362 |
| **Mobile:** | +44 (0) 7770326304 |
| **E-mail:** | d.l.giarettarl.ac.uk |
# CONTENT

1 **INTRODUCTION** ......................................................................................................................... 6  
1.1 PURPOSE OF THIS DOCUMENT .................................................................................................. 6  
1.2 CONCEPTUAL MODELS ................................................................................................................ 6  
1.3 RELATIONSHIP TO OAIS CONCEPTS .......................................................................................... 7  
1.4 HOW TO READ THIS DOCUMENT .............................................................................................. 8  
1.5 CASPAR OBJECTIVES .................................................................................................................. 8  
1.6 GLOSSARY .................................................................................................................................... 10  

2 **PRESERVATION CONCEPTS IN CASPAR** .................................................................................... 12  
2.1 FUTURE PROOFING IN THE REAL WORLD .................................................................................. 12  
2.1.1 Potential Points of failure ........................................................................................................ 13  
2.1.2 Potential Solutions .................................................................................................................. 13  
2.2 WHAT CAN BE RELIED ON IN THE LONG TERM? ...................................................................... 14  
2.3 LEVELS OF APPLICATION OF OAIS CONCEPTS ........................................................................ 14  
2.3.1 OAIS as a checklist .................................................................................................................. 14  
2.3.2 Preservation without automation .............................................................................................. 15  
2.3.3 Preservation with automation and interoperability ................................................................... 15  
2.4 ISSUES NOT COVERED IN DETAIL BY OAIS (OR CASPAR) ..................................................... 19  

3 **INFRASTRUCTURE CONCEPTS IN CASPAR** ............................................................................. 21  
3.1 MECHANISMS FOR SHARING THE BURDEN OF PRESERVATION ................................................ 21  

4 **KEY PRESERVATION COMPONENTS** .......................................................................................... 23  
4.1 DISCIPLINE INDEPENDENT ASPECTS ....................................................................................... 26  
4.1.1 Preservation Planning .............................................................................................................. 27  
4.1.1.1 Registries of Representation Information .............................................................................. 27  
4.1.1.2 Orchestration ........................................................................................................................ 28  
4.1.1.3 ReoInfo Gap Manager .......................................................................................................... 28  
4.1.2 Digital Object Storage ............................................................................................................. 29  
4.1.3 Ingest ....................................................................................................................................... 30  
4.1.4 Access ...................................................................................................................................... 30  
4.1.4.1 Access Control/DRM/Trust .................................................................................................... 30  
4.1.4.2 Finding Aids based on PDI ..................................................................................................... 31  
4.1.5 Data Management ................................................................................................................... 31  
4.2 DISCIPLINE DEPENDENCE: TOOLBOXES/LIBRARIES ............................................................ 31  

5 **KEY UNDERLYING TECHNOLOGIES** ......................................................................................... 33  
5.1 KNOWLEDGE MANAGEMENT AT THE HEART OF PRESERVATION ........................................... 33  
5.2 VIRTUALISATION ......................................................................................................................... 33  
5.2.1 Common Information Virtualisation ......................................................................................... 34  
5.2.1.1 Simple Objects ...................................................................................................................... 34  
5.2.1.2 Complex Objects .................................................................................................................. 38  
5.2.2 Discipline Specific Information Virtualisation ............................................................................ 38  
5.2.3 Higher Level Knowledge Virtualisation ...................................................................................... 39  
5.2.4 Access Control/Trust Virtualisation ............................................................................................ 40  
5.2.5 Digital Object Storage Virtualisation ......................................................................................... 41  

6 **REPRESENTATION INFORMATION** ............................................................................................ 43  
6.1 INTRODUCTION ............................................................................................................................ 43  
6.2 STRUCTURE ................................................................................................................................... 44  
6.3 SEMANTICS .................................................................................................................................... 44  
6.3.1 Technical Semantics .................................................................................................................. 45  
6.3.2 Intentional Semantics ................................................................................................................ 45  
6.4 OTHER ........................................................................................................................................... 45  
6.4.1 Behaviour, Actions and Processes ............................................................................................. 45  
6.4.1.1 Rights Enforcement ............................................................................................................... 45
1 INTRODUCTION

1.1 PURPOSE OF THIS DOCUMENT

This document is of fundamental importance to understanding the CASPAR project because it brings together the key concepts which shape CASPAR, namely those to do with digital preservation, guided by the OAIS Reference Model\(^\text{1}\), supplemented by the ideas of automation and discipline independence. The latter set of concepts helps us highlight those areas which can form a broadly applicable infrastructure to support digital preservation in a cost effective manner.

Another, but different, type of concept is that of recursion. This, appropriately enough, comes up time and again within CASPAR and through the Conceptual Model, and hence is worth noting here. By this we mean that components which appear at one level of granularity re-appear when we take a finer grained view, within the detailed breakdown of those or other components. This is familiar in mathematics but less so in other areas. As is well known in the former, it is important to understand where the recursion ends otherwise it becomes impossible to produce practical results. Examples include (anticipating OAIS terminology):

- Representation Information (RepInfo) itself needs its own Representation Information. The recursion stops at the Knowledge Base of the Designated Community.

- Any piece of Preservation Description Information (PDI) will itself be encoded as a Data Object, which needs Representation Information. Representation Information as a digital object within a Registry will also need its own PDI. The recursion in this case might end with a simple text description of PDI.

- Components of a preservation infrastructure themselves need to be preserved – for example a Registry which supports OAIS-type archives must itself be an OAIS-type archive in that it must be relied upon to preserve its (Representation) Information objects over the long term. The recursion may stop, for example, where a Registry contains sufficient RepInfo for its most basic information about its own holdings without needing to refer to external Registries. In general the functionality of the preservation components will need to be preserved and migrated into new systems so that the preservation support can continue into the future.

Finally we have the underlying concept of “proof” or “testability” of what is proposed. This is fundamental to CASPAR but addressed in the Testbed Scenarios (See D4101).

It is expected that some revisions of these concepts will arise as the projects moves into the implementation phase, and version 2 of this document will be released to reflect such changes.

1.2 CONCEPTUAL MODELS

Having described the purpose of this document it should be added that the notion of conceptual modelling is vague and ill-defined, with varying interpretations as to its meaning.\(^\text{2}\) However it is agreed that it involves an abstraction of a model from a real or proposed system which involves some degree of simplification.

\(^1\) [http://public.ccsds.org/publications/archive/650x0b1.pdf](http://public.ccsds.org/publications/archive/650x0b1.pdf)

\(^2\) [ISSUES IN CONCEPTUAL MODELLING FOR SIMULATION:SETTING A RESEARCH AGENDA](http://arthur.cs.vt.edu/conceptual-modelling/Manuscripts/SW06_20%20Robinson.pdf)
“An application's conceptual model is the "mental map" that the designers build into it in order to present information logically,..., the conceptual model represents how they expect and believe the information they are encountering should, and does, fit together."³.

It is generally agreed that these models must be simple, but must be sufficiently rich to be able to be evolved as a deeper understanding of what is needed is gained.

Principles of modelling have been proposed as²:

- "Model simple; think complicated"
- "Be parsimonious; start small and add"
- "Divide and conquer; avoid megamodels"
- "Use metaphors, analogies, and similarities"
- "Do not fall in love with data"

These principles have been adopted while developing the CASPAR Conceptual Model. The model is based on the CASPAR Description of Work [DoW], and aims to provide an overall view of the way in which we see preservation working. This document is provided primarily as a project-internal high level overview, and has been designed to (1) help the members of CASPAR project, and in particular Work Package leaders, ensure integration and consistency across the project and (2) help people outside the project understand the fundamental ideas behind the design. We expect the model to evolve as we understand the issues better. The Conceptual Model drives, and is consistent with, the Architecture, as the two documents are developed in close collaboration.

We have sought to avoid a number of obvious pitfalls, namely

1. being too much like an architecture – although the Architecture itself has to be consistent with the Conceptual Model, and
2. being focused too much on the here and now and forgetting about preservation, in particular forgetting about what may be needed by future technologies and future (as yet unborn) users of the digital information to be preserved.
3. Clearly we have no crystal ball to foresee the future but we believe that the fundamental concepts provided by OAIS allow us to develop processes to accommodate changes without knowing right now what those changes might be.

1.3 RELATIONSHIP TO OAIS CONCEPTS

Clearly the several models contained within the OAIS Reference Model¹ play important roles in this Conceptual Model and OAIS terminology will be used throughout. However it should be emphasised from the start that CASPAR is not implementing a turnkey archive system, which can be used to replace existing archives and hence some of the functional entities in the OAIS Functional Model will only be considered in passing. As a corollary, and as an obvious consequence of the time-limited funding of CASPAR, the project will not itself provide a long-term archive for digital information.

CASPAR seeks above all to provide generic infrastructure components whereas some of the OAIS functional entities, in particular INGEST, ACCESS and DATA MANAGEMENT, are very discipline specific. Furthermore the details of INGEST and ACCESS are, almost by definition, transient activities which will change project by project and even dataset by dataset. For these OAIS functional entities we focus on some fairly generic or novel features, although for the Testbeds we will have to provide some instance specific implementations of limited functionality, simply in order to complete the scenarios.

³ [http://www.air.org/concord/wai/conceptual.html](http://www.air.org/concord/wai/conceptual.html)
The focus on these infrastructure components is based on the belief that no single OAIS organisation can guarantee the resources required for long-term preservation - the effort must be shared, and the CASPAR infrastructure facilitates that process.

1.4 HOW TO READ THIS DOCUMENT

This document introduces concepts in what is believed to be a logical manner. In some cases concepts introduced at one point are elucidated in subsequent sections, after the introduction of the additional concepts which are required.

Section 2 discusses digital preservation within the “world-view” of OAIS, but also introduces a number of considerations which we believe must be used to supplement OAIS concepts in order to guide us towards generic infrastructure components, which are outlined in Section 3, and discussed in more detail in Section 4.

In Section 5 we look at some underlying technologies which are used throughout CASPAR. While CASPAR seeks to provide a number of generic infrastructure components, sections 6 and 7 discuss some of the discipline specific toolboxes which are necessary.

The design concepts which should be followed in order to try to ensure the long-term maintainability of the CASPAR implementations are discussed in section 9.

This document and D4101: User Requirements and Scenario Specifications [R2] should be read before going into the details of the Architecture, while the D1101: Review of the State of the Art provides details of many of the projects, tools and techniques which are mentioned in this document.

1.5 CASPAR OBJECTIVES

From the CASPAR Description of Work, the objectives of the project can be stated as follows:

<table>
<thead>
<tr>
<th>The CASPAR challenge is to achieve four main goals that can be stated as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal 1:</strong> build a pioneering preservation environment, based on a full use of the OAIS Reference Model' and building in the latest developments in knowledge technologies</td>
</tr>
<tr>
<td><strong>Goal 2:</strong> demonstrate its ability to handle the preservation of the digital resources of many user communities</td>
</tr>
<tr>
<td><strong>Goal 3:</strong> advance the current state of the art in digital preservation</td>
</tr>
<tr>
<td><strong>Goal 4:</strong> development of technological solutions supporting the emergence of an offer of systems and services for preservation of digital resources</td>
</tr>
</tbody>
</table>

Expanding these goals into more specific objectives, CASPAR will:

1. Implement, extend and validate the OAIS reference model.

2. Enhance the techniques for capturing Representation Information and other preservation related information for content objects.

3. Design virtualisation services supporting the preservation of digital resources over the long term, despite changes in the underlying computing (hardware and software) and storage systems, and the Designated Communities.

4. Integrate as standard features of CASPAR, digital rights management, authentication and accreditation.

5. Research more sophisticated access and use methods of preserved digital resources including intuitive query and browsing mechanisms.

6. Develop case studies demonstrating the validity of the CASPAR approach to the preservation of
digital resources across different user communities and assessing the conditions for a successful replication.

7. Actively contribute to the relevant standardisation activities in areas addressed by CASPAR.

8. Raise awareness about the critical importance of the preservation of digital resources among the relevant user-communities and facilitate the emergence of a more diverse offer of systems and services for preservation of digital resources.

Validation of these objectives is to be via a number of Measurable Objectives, addressing preservation aspects including:

- a sound theoretical basis and in particular alignment with the OAIS Reference Model,
- "accelerated lifetime" tests involving hardware, software and the knowledge base of the Designated Community,
- an increase in the trustworthiness of archives using CASPAR.
Applicable documents and reference documents

**Applicable documents**


[A2] Risk Form

**Reference documents**

[R1] CASPAR proposal, Sept 2005

[R2] D4101 USER REQUIREMENTS AND SCENARIO SPECIFICATIONS (CASPAR-D4101-SCEN-0101-1_0)

1.6 **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 PRESERVATION CONCEPTS IN CASPAR

There are a number of approaches to the complex issues in digital preservation, and many diverse aspects which need to be addressed, including social, financial, legal and technical considerations. While CASPAR may have some relevance to all these, the area in which we can have immediate impact involves technical issues, producing tools, techniques and procedures to support digital preservation.

The technical approach followed in CASPAR is guided by the Open Archival Information Systems (OAIS) Reference Model. OAIS is concerned with the long-term preservation of digitally encoded information, by which it means that the preserved data must continue to be understandable by a specified Designated Community. OAIS concepts are used throughout this document, but we expand on these concepts where appropriate.

As is well understood, it is also important in the lifecycle of a piece of digitally encoded information to decide whether it should be preserved and for how long. These appraisal and related issues are not addressed by CASPAR, however CASPAR should provide a better understanding of the implications of any decision about preservation.

It is worth first covering a number of more general ideas. These include some of the challenges to digital preservation and their possible solutions, some of the things on which we can rely in the long term. These considerations are followed by some discussion around OAIS and how its concepts might be used, which leads on to an introduction to some of the fundamental ideas which drive CASPAR. There is also an overview of those topics which OAIS (and CASPAR) pay only limited attention.

2.1 FUTURE PROOFING IN THE REAL WORLD

Within a single organisation, with a continuous supply of adequate funding, the job of digital preservation is at least feasible. However no-one can be sure of continued funding, and examples of such continued, and generous, funding are hard if not impossible to find. Instead the preservation of any piece of digitally encoded information almost certainly will rely on one or more chains of preservation each of which is only as strong its weakest link.

Example: a Space Agency which has decided that it will no longer provide funding for the preservation of a particular Earth Observation dataset may offer the data to other Space Agencies which may wish to continue the preservation of the dataset, thus continuing the chain of preservation.

In the following sub-sections we discuss some of the major potential points of failure in these chains and some of the ways in which these might be addressed. Subsequent sections provide more details of the concepts needed to support these solutions.
2.1.1 Potential Points of failure

Failure of any chain of preservation may be imagined as involving changes in, or non-maintainability of, essential hardware, software or support environment. Additionally the human methodology established for preservation may not be followed (sudden changes of a whole team of people, etc.)

OAIS stresses the importance of taking into account the changes in the Knowledge Base of the Designated Community. This may not be done adequately.

Additionally one may have a loss in the chain of evidence and lack of certainty of provenance or authenticity.

Encodings currently considered uncrackable, used to establish lack of tampering may eventually be broken using increasingly powerful processors or sophistication of attack.

The custodian of the data, an organisation or project, no matter how well established, may, at some point in the future, cease to exist.

Even if the organisation exists, the mechanisms to identify the location of data, for example a DNS entry pointing to a host machine, may no longer be resolvable.

Mandating the continued use of specific systems or formats is one possible way to try to ensure preservation. For example we might try to mandate all images to be JPEG, all documents to be PDF/A, and all science data to be kept as XML files, or demand that a specific ontology be adopted. Even if we were to be successful for a limited time, the one thing we can be sure of is that things would change and the mandates would fail.

2.1.2 Potential Solutions

One of the recognised techniques of isolating dependencies on hardware, software and environment is virtualisation. By this is meant the technique of identifying abstract, important, interfaces/processes which can be implemented on top of concrete implementations.

Changes in Knowledge Base can only be truly solved by the community itself, but procedures can be proposed which help to ensure that gaps in understandability are at least recognised and the information requested from the community before it is entirely lost.

Provenance and authenticity is, in part at least, dependent on social and information policy concerns, process documentation, and other aspects which cannot have a purely technical solution. However some tools can be made available to ameliorate the risks of security breaches. Systems security and data integrity are only two aspects of provenance and authenticity, and we should be careful not to assume that tools for these problems will provide solutions to larger problems.

Custodianship should always be regarded as a temporary trust and techniques are needed to allow a smooth handing over of holdings from one link in the chain of preservation to the next.

The provision of a definitive system of persistent actionable identifiers lies outside the remit of CASPAR however a system which spreads the risk of the deterioration of identifier systems can be proposed.

Given the constantly changing world we need a system which does not force a specific way of doing things but instead we should be able to allow anything to be accommodated. For example we cannot mandate a particular way of producing Representation Information. While it might have some advantages in terms of interoperability in the short term, in the long term we would be locked into a dead-end. However this should not prevent us from advising on best practise.
2.2 WHAT CAN BE RELIED ON IN THE LONG TERM?

While we cannot provide rigorous proofs, it is worth, at this point, listing those things which we might credibly argue would be available in the long term, in order to clarify the basis of our approach. We should be able to trace back our preservation plans to these assumptions. Were we able to undertake a rigorous mathematical proof these would form the basis of the axioms for our “theorems”.

- Words on paper (or titanium sheets) that people can read; ISO standards are an example of this. Over the long term there may be an issue of language and character shape.
  - Carvings in stone and books have proven track records of preserving information over hundreds of years.
- The information such as Representation Information which is collected.
  - A somewhat recursive assumption, however it is difficult to make progress without it. This Representation Information includes both digital as well as physical (e.g. books) objects.
- Some kind of remote access
  - Network access is the natural assumption but in principle other methods of obtaining information from a given address/location would suffice, for example fax or horseback rider.
- Some kind of computers
  - Perhaps not strictly necessary but this seems a sensible assumption given the amount of calculation needed to do some of the most trivial operations, such as displaying anything beyond simple ASCII text, or extracting information from large datasets.
- People? Organisations?
  - Clearly neither the originators of the digital objects nor the initial host organisations can be relied on to continue to exist. However if no people and no organisations exist at all then perhaps digital preservation becomes a moot topic.
- Identifiers?
  - Some kind of identifier system is needed, as discussed in sections 7.3.2.2, will be needed, but clearly we cannot assume that any given URL, for example, will remain valid.

With these in mind we can move on to some general considerations about future-proofing digitally encoded information.

2.3 LEVELS OF APPLICATION OF OAIS CONCEPTS

OAIS is not a design; its lack of specificity gives it wide applicability and great strength but it also forces implementers to make choices, among which is the level of application of the OAIS concepts.

2.3.1 OAIS as a checklist

OAIS “provides a framework, including terminology and concepts, for describing and comparing architectures and operations of existing and future archives.”

The simplest way of applying OAIS is as a checklist. In particular, instead of “do we have enough metadata?”, the question becomes “do we have Representation Information?, do we have Representation Information for that piece of Representation Information? do we have Preservation Description Information (PDI)?, do we have Packaging Information? etc”.

Similarly one can ask whether the various processes and functions defined in OAIS can be identified in an existing or planned archive.
2.3.2 Preservation without automation

Going beyond a simple checklist one can use OAIS as the framework for, for example, Representation Information. Here we must simply ensure that there is adequate Representation Information for the Designated Community. Other users may or may not be able to understand the data content.

Any piece of that Representation Information could itself be as “opaque” as any other piece of data. OAIS requires that each piece of Representation Information has its own Representation Information – with the recursion stopping where it meets, in a sense which needs to be properly defined, the Knowledge Base of the Designated Community – which itself needs to be adequately defined.

However even the Designated Community may need to put in a considerable effort, for example to read documentation and create specialised software at each level of the recursion, in order to understand and use the content.

Example: the Representation Information could be in the form of a detailed document describing, in simple text and diagrams, how the information is encoded. The text description would have to be read by a human and presumably software would have to be written – possibly requiring significant effort. The RFC system (http://www.ietf.org/rfc.html) is an example of this use of simple text files to describe all the major systems in the Internet.

2.3.3 Preservation with automation and interoperability

The next step is to try to ensure that the use of the Representation Information is as easy and automated as possible, and is widely usable beyond the Designated Community. This demands increasing automation in the access, interpretation and use of Representation Information, and also the provision of more clues to users from different disciplines.

For the latter one can begin by offering some common views on data – for example allowing easier use in generic applications – by means of virtualisation. An example of this would be where the information embodies an image, to make this fact explicit in the Representation Information so that an application would know that it makes sense to handle the data as a 2-dimensional image. In particular the data can be displayed; it has a size specified as a number of rows and columns. Further discussion is provided in section 5.2. This type of virtualisation is common in many other, non-preservation related, areas. It is the basis on which computer operating systems can work surviving many generations of changes in technologies, on a variety of hardware. For example, the operations which a disk drive must perform can be specified and used throughout the rest of the operating system, but the specifics of how that is implemented are isolated within a driver library. The underlying idea here is to define, in software terms, a set of interfaces which can be implemented on top of a variety of specific instances which will change over time.

We will apply these ideas of virtualisation even more widely, as illustrated in Figure 1.
In terms of access, interpretation and use of the Representation Information, the key concept here is to try to make the access to, and the form of, the initial piece of Representation Information as “standard” as possible. In CASPAR this piece of initial Representation Information is called the “RepInfoLabel” which will be described later. The purpose of this initial piece of RepInfo is to provide a categorisation of the types of RepInfo which are available for the Data Object, using the classification of RepInfo which OAIS provides. Such a breakdown gives users (and applications) a clue as to which piece of RepInfo is of relevance for any particular purpose.

Note that the CASPAR “RepInfoLabel” itself has Representation Information. The RepInfoLabel has been introduced for convenience, but is not in any sense unique or irreplaceable.
In terms of standardising the access, we propose that identifiers (called here **Curation Persistent Identifiers** - CPID) are associated with any data object, which point to the appropriate Representation Information, as illustrated in Figure 3. The concepts underlying these Persistent Identifiers are discussed in detail in section 7.3.2.2.

**Figure 2 OAIS Classification of Representation Information**

**Figure 3 Linking to Representation Information**

In this diagram we introduce the idea of a Registry/Repository of Representation Information. However it must be stressed that

- this is **not** intended to indicate a single central registry, which would be a single point of failure in such a preservation system, but rather a network of distributed, perhaps independent, registries and
the arrows are uni-directional, in other words there is a pointer from the “data” to its Representation Information BUT not necessarily vice-versa, because one piece of Representation Information might be applicable to many thousands of data instances.

The registry concept has the advantage that, as will be expanded on later in this document, it facilitates the sharing of the effort in producing Representation Information.

It must also be stressed that this conceptual model does not imply that all Representation Information is kept in Registries; in fact it is perfectly sensible to physically package Representation Information with the data content, in the Archival Information Package (AIP). However for any piece of information, changes in the knowledge base of the Designated Community imply that the amount of Representation Information which has been explicitly captured must change, and this is facilitated by being able to point outside of the AIP.

In order to tie this in with the idea of the initial piece of Representation Information, we can expand the first transaction as follows:

![Diagram of RepInfoLabel concept](Image)

**Figure 4 Use of RepInfoLabel**

The initial RepInfo (a RepInfoLabel) is circled in Figure 4; if the application needs some Semantic RepInfo, then the appropriate CPID is selected and the piece of RepInfo (something to do with Semantics) is obtained from the Registry/Repository and transferred back to the user. This piece of Semantic RepInfo may be understandable by the user; if not then it will itself have a CPID associated with it which points back to the Registry/Repository – to another RepInfoLabel. This iteration continues until the user can understand the RepInfo.

Another possible termination point is indicated by the CPID having the special value “MISSING”, which indicates that the Representation Information is not available – and this could signal that there is a RepInfo gap.

<table>
<thead>
<tr>
<th>Although not indicated, each RepInfoLabel also has a CPID which points to the Representation Information for that RepInfoLabel, which will not be another RepInfoLabel of the same type but instead will be a simple text file – in order to end the recursion.</th>
</tr>
</thead>
</table>

The above scenario describes the case where all transactions take place with a single Registry/Repository, but of course any CPID may point to any one of what may be a large network of
In terms of the getting to the point at which the Representation Information is adequate, this may be a human decision but some automation is possible.

Support for such automation is illustrated in Figure 5 which shows users (\(u_1, u_2\)…) with user profiles (\(p_1, p_2\)… – each a description of the user’s Knowledge Base) with Representation Information (\(m_1, m_2, m_3, m_4\)) to understand various digital objects (\(o_1, o_2\)…).

Take for example user \(u_1\) trying to understand digital object \(o_1\). To understand \(o_1\), Representation Information \(m_1\) is needed. The profile \(p_1\) shows that user \(u_1\) understands \(m_1\) (and therefore its dependencies \(m_2, m_3, m_4\)) and therefore has enough Representation Information to understand \(o_1\).

When user \(u_2\) tries to understand \(o_2\) we see that \(o_2\) needs Representation Information \(m_3\) and \(m_4\). Profile \(p_2\) shows that \(u_2\) understands \(m_2\) (and therefore \(m_3\)), however there is a gap, namely \(m_4\) which is required for \(u_2\) to understand \(o_2\).

For \(u_2\) to understand \(o_1\), we can see that Representation Information \(m_1\) and \(m_4\) need to be supplied.

2.4 ISSUES NOT COVERED IN DETAIL BY OAIS (OR CASPAR)

As noted at the start of this section OAIS notes but does not address in depth several issues, for example Authenticity does not properly cover the range of concerns of the many types of communities and this is one area which may be updated in the OAIS 5-year review. Other topics fell outside the remit of the OAIS standard; some of these were left for follow-on standards, while still others were thought to be too specialised to be amenable to this type of standardisation.

The former category includes:
- standard(s) for the interfaces between OAIS type archives;
- standard(s) for the submission (ingest) methodology used by an archive;
- standard(s) for the submission (ingest) of digital data sources to the archive;
– standard(s) for the delivery of digital sources from the archive;
– standard(s) for the submission of digital metadata, about digital or physical data sources, to the archive;
– standard(s) for the identification of digital sources within the archive;
– protocol standard(s) to search and retrieve metadata information about digital and physical data sources;
– standard(s) for media access allowing replacement of media management systems without having to rewrite the media;
– standard(s) for specific physical media;
– standard(s) for the migration of information across media and formats;
– standard(s) for recommended archival practices;
– standard(s) for accreditation of archives.

The latter category, namely those too archive/domain specific for OAIS-type standardisation includes:
– appraisal process for information to be archived
– finding aids
– access methods and Finding Aids
– details of Data Management
3 INFRASTRUCTURE CONCEPTS IN CASPAR

The concepts introduced here are aimed at providing an infrastructure which can be shared across the widest range of information. In that way we can spread the risk that the infrastructure upon which we would depend might itself disappear by broadening its potential support.

We begin by looking at those concepts which are discipline independent, and after that we identify where the domain dependence occurs, as it must.

We further focus on Representation Information in that that can be shared between data archives, whereas the Preservation Description Information (PDI) tends by its nature to be unique to data instances and archives.

Some of the components which we identify, illustrated in Figure 6, could conceivably be large, unique, resources, for example the Registry and the RepInfo Gap Manager (although issues of scalability would naturally arise). Other components make no real sense as single resources – for example the Archival Store (i.e. it would be difficult to advocate keeping all data in a large central data store). Nevertheless it would make more sense to allow multiple instances of any component, in order to provide adequate resilience and scalability. For example it would be perfectly reasonable to have multiple independent Registries, some of which may interoperate or federate.

3.1 MECHANISMS FOR SHARING THE BURDEN OF PRESERVATION

The rather generic scenario which Figure 6 captures is as follows:

- A user uses an application to perform some task – this can range from displaying a document to performing some scientific analysis to playing a piece of music.
- The user must find the appropriate data needed for the task (using Finding Aids), then must obtain the data (from the Archive)
- Either the user knows how to use the application to deal with the data or else, by definition, additional Representation Information is needed.
- Additional Representation Information may be obtained from the Archive, or else from somewhere else (from the Registry). That Representation Information may itself need further Representation Information.

In the future, as time passes, additional Representation Information will be needed, as the Knowledge Base of the Designated Community changes. The question is: how can this Representation Information be identified and generated? The scenario we have in mind, bearing in mind the previous discussion, is the following:

- People know that things have changed – this may be the Data Preservers, or a member of the Designated Community. These people inform something – the RepInfo Gap Manager.
- The RepInfo Gap Manager identifies a gap in some Representation Information net. It is unlikely that the RepInfo Gap Manager itself can fill this gap. In general only people can fill the gap, by generating new Representation Information. The next question is ”who should be asked to do this?” Again, the only answer to this has to be provided by the individuals themselves.
- In order to facilitate this network of people we need a component (the Orchestration Manager) where people can register their experience or expertise. The RepInfo Gap Manager informs the Orchestration Manager that some type of Representation Information is needed. The Orchestration Manager sends requests to those with the appropriate expertise.
- If such a person can be found then they can generate the Representation Information using anything at hand, but some set of tools would be useful (Representation Information Toolkit). The newly created Representation Information is placed somewhere (most likely in a Registry)
The claim is that, apart from the Representation Information toolkit, the other components mentioned above are not discipline dependent, and this is underpinned by the fact that the Representation Information is, as far as those components are concerned, opaque binary objects.

![Figure 6 Key Infrastructure Components](image-url)
4 KEY PRESERVATION COMPONENTS

As argued above, we can construct a discipline independent infrastructure to support the maintenance of Representation Information. The complexity of the Representation Information itself is has been isolated in the Representation Information toolkit.

In this section we look in more detail at the Representation Information concepts and associated components, and also at the other metadata which must be captured in the preservation process.

The following diagram and table pick out the main components, following the lifecycle of a piece of digitally encoded information as it is ingested into an OAIS system and subsequently retrieved for use at some time in the future.

![Image](CASPAR Information Flow Architecture)

**Figure 7 CASPAR Information Flow Architecture**

<table>
<thead>
<tr>
<th>Step in Lifecycle</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create the Information Object – by adding Representation Information to the particular digital data object.</td>
<td>Guides the user to a number of applications to create adequate Representation Information (RepInfo) – using existing RepInfo from registries where available. Representation Information includes Syntactical and Semantic descriptions and also associated software and standards.</td>
</tr>
<tr>
<td>1. Representation Information – adequate for the Designated Community – must be created. <strong>COMPONENT:</strong> Representation Information Toolbox</td>
<td></td>
</tr>
<tr>
<td>2. Capture Data Management information</td>
<td>This includes Access Control information such as Access Control Lists. More detailed DRM may also be needed.</td>
</tr>
<tr>
<td>COMPONENT: Data Access Manager</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>3. Capture Digital Rights associated with the data object</td>
<td></td>
</tr>
</tbody>
</table>

Digital Rights Management can be an essential part of digital preservation, especially for the short- to medium-term portion of archival storage.

<table>
<thead>
<tr>
<th>COMPONENT: DRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Capture Higher level Semantics</td>
</tr>
<tr>
<td>COMPONENT: RepInfo Gap Manager</td>
</tr>
</tbody>
</table>

The capture of higher-level, more subtle semantics (knowledge) is closely related to that which the burgeoning Semantic Web research industry is working on. What is needed here is to ensure that what is captured can survive over the long term and over many changes in “knowledge technologies”. The RepInfo Gap Manager is applicable at all levels of description, but the most challenging is the more complex, higher-level, semantics.

<table>
<thead>
<tr>
<th>COMPONENT: Virtualisation Assistant</th>
</tr>
</thead>
<tbody>
<tr>
<td>5. Create Virtualisation description</td>
</tr>
</tbody>
</table>

Virtualisation consists of identifying abstractions – probably many different types of abstractions – to encapsulate important features of the Information Object.

This is essentially OAIS Representation Information, but we wish to stress the Virtualisation aspects because of the need to facilitate automated processing.

<table>
<thead>
<tr>
<th>In the taxonomy of Information Objects, it is useful to distinguish between Simple and Complex Objects, as well as discipline-specific virtualisation, hence the need for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>6. Describe discipline specific object characteristics</td>
</tr>
<tr>
<td>COMPONENT: Discipline specific Object Virtualiser Manager</td>
</tr>
</tbody>
</table>

In order to limit the multiplicity of types of object, it seems reasonable to normalise characteristics, separating those that are discipline specific from those that are common to all objects.

<table>
<thead>
<tr>
<th>COMPONENT: Simple Object Virtualiser</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Describe Simple Object</td>
</tr>
</tbody>
</table>

The “simple” in this case refers to the type of Information Object; this is a non-trivial component involving descriptions of the Structure as well as the Semantics of relatively self-contained Objects.

<table>
<thead>
<tr>
<th>COMPONENT: Complex Object Virtualiser</th>
</tr>
</thead>
<tbody>
<tr>
<td>8. Describe Complex Object</td>
</tr>
</tbody>
</table>

A Complex Object is one that can be described in terms of several, possibly a large number of, both Simple Objects and Complex Objects and their inter-relations. In particular, the Complex Object Virtualiser has to cope with multiple partial copies of the datasets forming the referred Object and with the management of the lower-level objects in a fully distributed environment.

<table>
<thead>
<tr>
<th>COMPONENT: On-Demand Object Virtualiser</th>
</tr>
</thead>
<tbody>
<tr>
<td>9. Some of the Objects are created “on-demand”. It may be true to say that most Information is created in this way.</td>
</tr>
</tbody>
</table>

An On-Demand Object is one (Simple or Complex) that can be referred to by the available knowledge and can be instantiated on request.

<table>
<thead>
<tr>
<th>COMPONENT: Preservation Description Information (PDI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. Produce OAIS Preservation Description Information (PDI)</td>
</tr>
</tbody>
</table>

PDI includes Fixity, Provenance, Reference and Context information. The PDI Toolbox has a number of sub-components that address each of these and guide the user to produce the most complete PDI possible. Knowledge capture techniques should also be applicable here.
Having created the Virtualisation information (which includes and extends the OAIS concept of Representation Information), it must be stored in an accessible location.

11. Store DRM/Virtualisation Information/ Representation Information in Registry

**COMPONENT:** Registry

We take the general case that it is stored in an external registry in order to allow the possibility of enhancing the Representation Information etc. to cope with changes in the technologies, Designated Communities etc.

The alternative of storing this metadata with the data object is possible but would not address long-term preservation because:

- the RepInfo cannot be complete;
- the RepInfo cannot easily be updated;
- the RepInfo has to be repeated for each object or copy of the object and consistency cannot easily be maintained;
- the effort of updating the RepInfo cannot easily be shared, in particular when the originator is no longer available.

In addition to Representation Information, there may also be keys, public and private, for encryption etc. that need to be available over the long term.

12. Store Keys, public and private

**COMPONENT:** Key Store

Public keys can be stored in any convenient location that is accessible to users. However, for long-term preservation these keys must be guaranteed to be available, as must the appropriate encryption or digest algorithm.

The same applies for private keys, which must be held in “escrow” for some agreed period, with adequate security.

The collection of information adequate for preservation is a key concept in OAIS – the Archival Information Package.

13. Construct the AIP

**COMPONENT:** Archival Information Packager

The AIP is a logical construct, and key to preservation in the OAIS Reference Model. The AI Packager logically binds together the information required to preserve the Content Information so that it is suitable for long-term preservation. However, this should not be regarded as a static construct, since, as has been stressed, preservation is a dynamic process. The AI Packager works with the Preservation Orchestration Manager.

Having the AIP, this must now be securely stored.

14. Store the AIP data object securely for the long term.

**COMPONENT:** Preservation Data Object (PDO)

Digital storage comes in many different forms, and the hardware and software technology is constantly evolving. The PDO virtualises the storage at the level of a data object; in this way, it extends the current virtualised storage, which allows transparent access to distributed data. The PDO hides the details of the storage system, the collection management etc. – all of which can cause a great deal of trouble when migrating, as hardware and software technology changes.

One way of looking at this is to view it as an implementation of the OAIS Archival Storage functional element. As such, it allows the development of a market of interchangeable “Archival Storage” elements for a variety of archives.

Now we come to the period when the data object is stored for many years – in principle indefinitely. During this time the originators of the data pass away; hardware and software become obsolete and are replaced; the organisation that hosts the repository evolves, merges, perhaps terminates (but hands on
its data holdings); the community of users, their tools, their underlying Knowledge Base change out of all recognition.

In the background, something must keep the information alive, in the same way as the body’s autonomic nervous system keeps the body alive, namely by triggering breathing, heartbeat etc. Note that the autonomic nervous system does not actually do the breathing etc., but provides the trigger.

This is what must be arranged.

15. Notify the repository when changes must be made

**COMPONENTS:**
- ReplInfo Gap Manager
- Preservation Orchestration Manager

This will provide a number of notification services to alert repositories, which have registered appropriately, of the probable need to take action to ensure the preservation of their holdings. This action could range from the need for migration to new formats to the obsolescence of hardware to the availability of relevant Representation Information. In addition, brokering services and workflow control processes will be available to assist data holders to access services – for example, to transform data or to hand over holdings to longer-lived repositories.

Activities include advising on preservation strategies, providing support for Preservation Planning in repositories, and sharing Representation Information.

Without this type of background, activity preservation is at risk by neglect. Clearly, larger organisations may not need this, but, even in the largest and best run organisations, individual preservation projects may be funded on a relatively short-term basis.

This infrastructure must itself be persistent.

---

**4.1 DISCIPLINE INDEPENDENT ASPECTS**

The OAIS Functional Entities in the Functional Model can be used to group the domain independent concepts and components.

![Figure 8 OAIS Functional Model](image-url)

IST-2006-033572 PUBLIC 26 / 53
4.1.1 Preservation Planning

4.1.1.1 Registries of Representation Information

The Registry/Repository concept was introduced in section 2.3.3— the term Registry/Repository is used, rather than simply “Registry”, in order to stress the fact that the concept embodies the holding (in the Repository) of a significant amount of digital information – the Representation Information – rather than simply pointers to external resources.

The prime functions of a Registry/Repository are:

- Given an identifier of a piece of Representation Information (RepInfo), return that piece of Representation Information to the requestor. This Representation Information will in general be an opaque binary object as far as the Registry/Repository is concerned.
- Allow searching of the holdings of the Registry in order to enable the re-use of existing RepInfo.
  - To facilitate this searching, each piece of RepInfo should be classified under one or more Classification Schemes, and have a searchable text description of the RepInfo.
- Each piece of RepInfo should itself have a pointer to its own RepInfo, and also details of its PDI.
- The Registry/Repository should itself be an OAIS which can be certified for long-term preservation of information.

The basic functionality required of the Registry/Repository has been shown in Figure 3 and Figure 4 with the associated discussion in section 2.3.3.

The Registry/Repository functionality is domain independent because pieces of the Representation Information are, as far as the Registry/Repository is concerned, opaque binary objects.

Of course any piece of Representation Information could be domain specific, but that content is not relevant to the Registry/Repository. It is important to note that there may be multiple ways to describe something. For example Structure-type Representation Information may come in the form of an EAST description, or a DRB description or a DFDL description (further details of which may be found in D1101 – Review of State of the Art). All these are valid and each of these will have their own Representation Information.

In addition, it is possible that two archives may have identical copies of a piece of data but may provide entirely separate pieces of Representation Information. This is in many ways a duplication of effort. However the Registry/Repository will be entirely unaware of this duplication since (1) it does not have a link back to the data, as this would be unmaintainable and (2) the pieces of Representation Information are opaque binary objects as far as it is concerned.

A separate, value added, service may be developed by analysing the links between data and Representation Information, in a way analogous to the page ranking algorithm used by Google. Such a service would enable one to say, for example, that 99% of all archives use CPIDYYYY as the Representation Information for a certain type of data. Such a statistic may influence others to use that particular piece of Representation Information rather than some other, competing, Representation Information.

New versions of Representation may be created from time to time, to improve usability or accuracy. The versioning must be controlled and it will prove useful to distinguish between a unique identifier for a particular version and a logical identifier for all versions of the Representation Information. Using the logical identifier should return the latest (and presumably the best) version, which will change as new versions are created, whereas using the unique identifier, or, equivalently, providing a specific version number, should always provide that specific piece of Representation Information.

Representation Information may be cached, that is to say copies may, for convenience, be kept, in a variety of locations, including packaged with a data object. Caching is a well known optimisation technique and the appropriate steps must be taken to ensure that the cache copies are identical with the
original, however the task is made easier because a particular piece of Representation Information is never changed, instead, as discussed above, a new version is created.

### 4.1.1.2 Orchestration

The Orchestration component has to:

- allow individuals to register their interests and expertise
- collect information from (anonymous or registered) individuals about changes in software, hardware, environment or Knowledge Base of any Designated Community. This information will be passed on to the RepInfo Gap Manager component.
- receive information from the RepInfo Gap Manager component about a gap which has been identified
- send requests to appropriate registered users, based on their interests and expertise, for the creation of required Representation Information

The Orchestration functionality is domain independent in that it needs no embedded domain specific knowledge in order to match keywords specifying gaps to people, although clearly some domain specific thesauri could help give a wider set of relevant matches.

### 4.1.1.3 RepInfo Gap Manager

The RepInfo Gap Manager component embodies a small but essential application of Knowledge Management techniques to preservation. Its main purpose is to assist in identifying gaps which have arisen as a result of changes in hardware, software, environment and Knowledge Bases of Designated Communities.

The changes are notified by human participants in the preservation process. The RepInfo Gap Manager knows of the existing dependencies between pieces of Representation Information, working closely with one or more instances of the Registry/Repository. The labels in the Registry/Repository capture those dependencies. The changes imply that gaps in the Representation Information network will have arisen, which must be filled. Human participants must be alerted and requested to provide new Representation Information to fill those gaps. The human participation may not always be necessary; the RepInfo Gap Manager may be able to bring in Representation Information from another, existing, source to fill the gap – although this would have to be checked by humans.

As an example of these gaps we can look at the dependencies in the Representation Information about a piece of astronomical data. FITS is a standard data format that is used in astronomy. To understand a FITS file one needs to understand the FITS standard which is in turn described in a PDF document. To understand the keywords contained in a FITS file one needs to be able to understand the FITS dictionary (that explains the usage of keywords). Figure 9 illustrates these dependencies.

![Figure 9 FITS file dependencies](image)

**Figure 9 FITS file dependencies**
At some particular point in time the Dictionary may be part of the Knowledge Base of the Designated Community (i.e. astronomers). However there may come a time when this particular type of Dictionary begins to fall from general use. A gap in the Representation Information network will begin to appear, which must be filled. In most cases some human participant will have to create the additional piece of Representation Information that is required. However it may be the case that in some separate Representation Information network uses the same Dictionary and provides Representation Information for the Dictionary. The RepInfo Gap Manager may be able to deduce that the latter can be re-used in the astronomical case.

The RepInfo Gap Manager manipulates symbols and identifiers and does not require embedded domain specific knowledge.

4.1.2 Digital Object Storage

The Digital Object Storage (or sometimes simply “Storage”) component takes care of the “Digital Object” and encapsulates:

- The secure preservation of the bits which encode the information of interest. This of course applies to a primary Data Object, Representation Information, Preservation Description Information etc., the latter also being Data Objects. These individual stored objects form the simplest element in the storage system, and each needs only be regarded as opaque binary objects, whose internal structure need not be known or understood by the Storage system, although the structure of the AIP, e.g. how to get the PDI object out of the AIP, will be known to it.

- The association of Representation Information and PDI with the Content Information. This association may include having copies of the Representation Information or PDI kept within the Storage system. However it is important to recognise that neither of these can be complete. For example the Representation Information Network will change as, for example, the Knowledge Base of the Designated Community changes. Similarly the Provenance information will include not just the technical information about copying but also but also include descriptions of various real-world entities (eg. persons, organisations and their attributes, roles and actions) whose social context is also associated with the data. Therefore both Representation Information and PDI will have to include a pointer out of the storage system.

- The automatic maintenance of the technical provenance information, including details of what are essentially internal events including copying, replication and refreshment and the objects.

- The policies which the archive imposes on the stored objects (and the Representation Information, PDI etc associated with the encoded instances of these policies), for example
  - the number of backup copies, offsite and on-site, on-line and near-line, and replication
  - the access controls
  - the distribution of information among the individual pieces of virtualised storage
  - maintenance of namespaces
  - maintenance of collection level information

- The ability to hand on the stored AIPs, and appropriate collection information, to another OAIS system – either because of technological change or because of organisational change as the preserved information is passed on to the next in the chain of preservation.

The Digital Object Storage concept is intrinsically domain independent.
4.1.3 Ingest

The INGEST functional entity in the OAIS Reference Model provides

the services and functions to accept Submission Information Packages (SIPs) from Producers (or from internal elements under the OAIS Administration control) and prepare the contents for storage and management within the archive. Ingest functions include receiving SIPs, performing quality assurance on SIPs, generating an Archival Information Package (AIP) which complies with the archive’s data formatting and documentation standards, extracting Descriptive Information from the AIPs for inclusion in the archive database, and coordinating updates to Archival Storage and Data Management.

The OAIS Producer-Archive Interface Methodology Abstract Standard (PAIMAS) seeks to identify, define and provide structure to the relationships and interactions between an information Producer and an Archive. It defines the methodology for the structure of actions that are required from the initial time of contact between the Producer and the Archive until the objects of information are received and validated by the Archive. These actions cover the first stage of the Ingest Process. It is expected that a specific standard or ‘community standard’ would be created in order to take into account all of the specific features of the community in question.

The Producer-Archive Interface Specification (PAIS - a draft standard at the time of writing) aims to provide a standard method to formally define the digital information objects to be transferred by an information Producer to an Archive and for effectively transferring these objects in the form of SIPs.

Both PAIMAS and PAIS define a number of general concepts and provide check-lists against which the INGEST process can be judged.

The general concepts and checklists provided by PAIMAS and PAIS provide domain independent views of the processes that are needed in INGEST.

4.1.4 Access

ACCESS is the OAIS functional entity which provides

the services and functions that support Consumers in determining the existence, description, location and availability of information stored in the OAIS, and allowing Consumers to request and receive information products. Access functions include communicating with Consumers to receive requests, applying controls to limit access to specially protected information, coordinating the execution of requests to successful completion, generating responses (Dissemination Information Packages, result sets, reports) and delivering the responses to Consumers.

Looking at existing archives one sees a very great variety of ACCESS-type functions. Indeed it is probably true to say that this, the user-facing part of an archive’s work, is the area in which the archive will seek to “brand” its services. Clearly the access services have a certain degree of standardisation to allow interoperability, examples of which include provision of Web pages, OMI-PH harvesting, and FTP services. Nevertheless each archive will seek to provide a richer set of “branded” ordering, searching and data provision services, and thus there are limits to the type of domain independent services which CASPAR might offer to any archive.

Areas in which we might hope for some discipline independence are Access Control and specialised Finding Aids based on PDI, and these are considered next.

4.1.4.1 Access Control/DRM/Trust

Access Control, Trust and Digital Rights Management must attempt to withstand changes in:

- individuals, and their roles and even their existence

http://public.ccsds.org/publications/archive/651x0b1.pdf
A digital object may be deposited in an archive with one particular system of Access controls and DRM, but may (in fact certainly will) be used under a completely different access control system.

While DRM systems could be made specific to domains, the requirement for survivability to change will tend to require a significant independence from domain considerations.

### 4.1.4.2 Finding Aids based on PDI

A Finding Aid is defined in OAIS as a software program or document that allows Consumers to search for and identify Archival Information Packages of interest.

If the Consumer does not a priori what specific holdings of the OAIS are of interest, the Consumer will establish a Search Session with the OAIS. During this Search Session the Consumer will use the OAIS Finding Aids that operate on Descriptive Information, or in some cases on the AIPs themselves, to identify and investigate potential holdings of interest. This may be accomplished by the submission of queries and the return of result sets to the Consumer.

OAIS provides terminology for the information which is used by the Finding Aids, for example Descriptive Information, Associated Descriptions and Collection Descriptions. However further specification of this information is not provided by OAIS, in part because of the great variety of types of information which could be involved.

A type of Finding Aid which could have some discipline independent aspects is based on standardised PDI components, and in particular discipline independent aspects of Provenance, which CASPAR will be investigating in detail.

### 4.1.5 Data Management

The DATA MANAGEMENT OAIS functional entity is the entity that contains the services and functions for populating, maintaining, and accessing a wide variety of information. Some examples of this information are catalogs and inventories on what may be retrieved from Archival Storage, processing algorithms that may be run on retrieved data, Consumer access statistics, Consumer billing, Event Based Orders, security controls, and OAIS schedules, policies, and procedures.

Descriptive Information, mentioned above, is the set of information, consisting primarily of Package Descriptions, which is provided to Data Management to support the finding, ordering, and retrieving of OAIS information holdings by Consumers.

While in general this type of information is extremely diverse, there are some inventory activities which seem particularly basic and which requires relatively straightforward collection of information.

This domain independent type of Descriptive Information, used by the Data Management entity, is the simple catalogue of which Content Information is in which Archival Information Package, and this can be provided by CASPAR, at least to support the testbed scenarios.

### 4.2 DISCIPLINE DEPENDENCE: TOOLBOXES/LIBRARIES

As noted in the description of the Registry/Repository, individual pieces of Representation Information are opaque binary objects to it. However the Representation Information must contain specific information about some specific data objects, and must include discipline dependence.
The discipline specificity is captured using a variety of tools and techniques; the umbrella term “toolbox” includes all of these. Section 6 provides an overview of the types of Representation Information.

Discipline specificity is also needed for parts of the Preservation Description Information (PDI), and an umbrella toolbox is needed here also. PDI is discussed in more detail in section 7.

The term toolbox should not be interpreted as a Graphical User Interface (GUI), rather is just an umbrella term which could include, for example, many GUIs, software libraries, processes and procedures. There are a number of technologies which appear in many different guises, and these are described in section 5.
5 KEY UNDERLYING TECHNOLOGIES

5.1 KNOWLEDGE MANAGEMENT AT THE HEART OF PRESERVATION

Knowledge Management is used in various ways in digital preservation, including:

- Semantic Representation Information (captured in components in the Representation Information Toolbox)
- Enabling the definition of, and the tracking of changes in, Knowledge Bases of Designated Communities, as well as changes in hardware, software and environment (in the RepInfo Gap Manager)
- Evolving the virtualisation descriptions across time by means of tracking terminology evolution (threaded throughout the other components).

These areas will be discussed separately in the following sections.

5.2 VIRTUALISATION

Virtualisation is a term used in many areas. The common theme of all virtualisation technologies is the hiding of technical detail, through encapsulation. Virtualisation creates external interfaces that hide an underlying implementation. The benefits for preservation arise from the hiding of the specific, changing, technologies from the higher level applications which use them.

The Warwick Workshop\(^5\) noted that Virtualisation is an underlying theme, with a layering model illustrated in Figure 10.

![Virtualisation layering model](image)

Figure 10 Virtualisation layering model

However virtualisation is not a magic bullet. It cannot be expected to be applied everywhere, and even where it can be applied the interfaces can themselves become obsolete and will eventually have to be re-engineered/re-virtualised, nevertheless we believe that it is a valuable concept.

Each of these levels of virtualisation will have its own type of “virtualisation description”, which is a type of Representation Information, which will also need its own Representation Information.

The Wikipedia entry for Virtualisation\(^6\) provides an extensive list of types of virtualisation, and distinguishes between

- Platform virtualisation, which involves the simulation of virtual machines.


• Resource virtualisation, which involves the simulation of combined, fragmented, or simplified resources.

In Figure 1 indicates in somewhat more detail than Figure 10 a number of layers in which we expect to use Virtualisation including:
• Digital Object Storage virtualisation – discussed in section 5.2.5.
• Common information virtualisation
• Discipline specific information virtualisation

Virtualisation also applies to the
• Higher level knowledge
• Access control
• Processes

Of course even the Persistent Preservation Infrastructure has to be virtualised.

Each of these is discussed in more detail below, introducing the various concepts in a logical manner, for simplicity, which however does not follow the layering schemes in Figure 1 or Figure 10.

5.2.1 Common Information Virtualisation

The Common Information Virtualisation envisaged in CASPAR tries to extract those properties of an Information Object which are widely applicable.

5.2.1.1 Simple Objects

There are several types of relatively simple objects which appear again and again in scientific data, including images, trees, tables and documents. The benefit of this type of virtualisation is that for each of them one can rely upon a certain – admittedly simple – behaviours which are rather familiar to many software applications.

In software terms these virtualisations would be regarded as data types which have an associated API. The specialisations would each support the parent API but add new methods or interfaces. This is a common approach in Object Oriented programming and some references to existing software libraries are provided where appropriate.

Many of these software libraries provide a great deal of functionality built on top of a small core set of interfaces which must be implemented for a new implementation. It is this core set of interfaces which are of particular interest in CASPAR because the other capabilities can be built on top of them. Identifying this small core set of functions means that if we can indicate how to implement these for a piece of data then, right now, we can use rich sets of software applications, and in the future we have the core capabilities which stand a good chance of being implemented in future software systems.

5.2.1.1.1 Images

In common usage, an image or picture is an artefact that reproduces the likeness of some subject—usually a physical object or a person7. An image may be thought of as a digital object which may be displayed as a rectangular 2-dimensional array in which all the picture elements (pixels) have the same data type, and normally any two neighbouring pixels have some type of mathematical or physical relationship e.g. they help to make up a part of a picture. All 2-dimensional images have a number of common features, including

• Size
  o number of rows

---

The most basic kind of table. Certain considerations follow from this simplified description:

- number of columns i.e. all rows have the same number of pixels, making a rectangular array
  - Pixel type – same for all pixels
  - Attributes (name, value pairs)

The digital encoding of the image may not be a simple rectangular array of numbers – there may be compression for example. Such encoding are not of concern in this virtualisation. The same image may have many different digital encodings, each of which needs some appropriate Structural Representation Information.

The Java2D\(^8\) and the java.awt.Image\(^9\) provide sets of interfaces with a very rich set of capabilities for manipulating graphics and images. The java.awt.Image has a core set of methods which match the above list, namely getHeight, getWidth, getSource and getProperty.

Put into a wider context one can view images as a special case of n-dimensional arrays of data as follows:

### Figure 11 Image data hierarchy

5.2.1.1.2 Tables

A table\(^10\) consists of an ordered arrangement of rows and columns. This is a simplified description of the most basic kind of table. Certain considerations follow from this simplified description:

- the term row has several common synonyms (e.g., record, k-tuple, n-tuple, vector);
- the term column has several common synonyms (e.g., field, parameter, property, attribute);
- a column is usually identified by a name;
- a column name can consist of a word, phrase or a numerical index;

The elements of a table may be grouped, segmented, or arranged in many different ways, and even nested recursively. Additionally, a table may include metadata, annotations, header, footer or other ancillary features.

---


\(^{9}\) [http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Image.html](http://java.sun.com/j2se/1.4.2/docs/api/java/awt/Image.html)

\(^{10}\) [http://en.wikipedia.org/wiki/Table_%28information%29](http://en.wikipedia.org/wiki/Table_%28information%29) viewed 10 April 2007
Tables can be viewed as columns of information – each column has the same type – as illustrated in Figure 12 which comes from the Starlink Tables Infrastructure Library (STIL) table interface, which is rather rich in functionality and which is itself built on top of the Java TableModel interface. The latter has a core set of methods, namely:

- get the number of columns (getColumnCount)
- get the column names (getColumnName)
- get the number of rows (getRowCount)
- get the value at a particular cell (getValueAt)

![Figure 12 Example Table interface](image)

As with images, this may be viewed in a hierarchy as follows:

---

11 [http://www.starlink.ac.uk/stil](http://www.starlink.ac.uk/stil)
12 [http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/table/TableModel.html](http://java.sun.com/j2se/1.4.2/docs/api/javax/swing/table/TableModel.html)
5.2.1.3 Trees

In computer terms a tree is a data structure tree that emulates a tree structure with a set of linked nodes, each of which has a single parent node – except the (single) root node – and there are no closed “loop” structures (i.e. it is acyclic). A node with no children is a “leaf” node. This type of structure is illustrated in Figure 14, and it appears in many areas including XML structures. A variety of tree structures can be created by associating different properties with the nodes.

5.2.1.4 Documents

Simple documents, i.e. something with text and images that can be displayed to a user, can also be virtualised; an example of this is the Multivalent Browser¹³, which defines common access methods to


---

**Figure 13 Table hierarchy**

**Figure 14 Tree structure**

---

**TableData**

- **TimeSeries**
- **Catalogue**

- **Object Catalogue**
- **Observation Log**

---

5.2.1.3 Trees

In computer terms a tree is a data structure tree that emulates a tree structure with a set of linked nodes, each of which has a single parent node – except the (single) root node – and there are no closed “loop” structures (i.e. it is acyclic). A node with no children is a “leaf” node. This type of structure is illustrated in Figure 14, and it appears in many areas including XML structures. A variety of tree structures can be created by associating different properties with the nodes.

5.2.1.4 Documents

Simple documents, i.e. something with text and images that can be displayed to a user, can also be virtualised; an example of this is the Multivalent Browser¹³, which defines common access methods to

documents in a number of formats including scanned paper, HTML, UNIX manual pages, TeX DVI and PDF. The Multivalent browser central data structure is the document tree – a specialised version of the tree structure described in section 5.2.1.1.3.

Another, simpler, document model is provided by the W3C’s Document Object Model (DOM)\textsuperscript{14}.

5.2.1.1.5 Primitive data types
In addition to the fairly sophisticated simple objects, it is worth noting that most programming languages also have primitive data type including

- Numerical – integer, floating point, etc
- String – character based
- Boolean – true/false

5.2.1.2 Complex Objects
The concept Complex Object is, in CASPAR, a catch-all term which covers a variety of structured (tree-like) objects, which may contain other complex and simple objects.

The boundary between Simple Objects and Complex Objects is not sharp. For example Tree-type where the leave nodes are not primitive types may be considered a Complex Object; the Multivalent Browser document model may be rather complex. Nevertheless it is worth maintaining the distinction between

- Simple Objects, where we have some chance of being able to do something sensible with the information content using widely applicable, reasonably standard, interfaces – display, search, process etc.

and

- Complex Objects, which are likely to require a number of additional steps to unpack the individual Simple Objects – however the difficulty is then that the relationship between those Simple Objects has to be defined elsewhere. Usually creators of Complex Objects embed the knowledge of those relationships within associated software. These relationships may be captured using Knowledge Management techniques.

5.2.1.2.1 On-demand Objects
In the process of managing objects and creating, for example, DIPS, there is a need to create objects “on-the-fly”. One can in fact regard on-demand as the norm, depending on the level of detail at which one looks at the systems; there are many processes hidden from view in the various hardware and software systems.

Of more immediate interest are processes and workflows which act on the data objects to produce some desired output.

There are a variety of workflow description languages and types of process. The virtualisation required here is an abstract layer which can accommodate several different underlying workflow systems.

5.2.2 Discipline Specific Information Virtualisation
As noted above, each of the common virtualisations in the previous section is useful because one can rely on some (simple) specific behaviour from each type. Although simple the behaviours can be combined to produce quite complex results. However different disciplines can produce a number of specialised types of, for example, images. By this is meant that a number of additional, specialised, behaviours become available for each specialised type.

\textsuperscript{14} http://www.w3.org/DOM/
Figure 15 shows some examples of specialisations of image types. The Astronomical image will add the functionality of, for example, a World Coordinate System i.e. the Right Ascension/Declination of object at the centre of the image, and the direction and angular size on the sky of each pixel in the image. The set of FITS\textsuperscript{15} image standards provide the basis of this type of additional functionality.

Astronomical images can in turn be specialised further so that, for example, an X-Ray image can add the functionality of providing the energy of each X-ray photon collected by the observing instrument.

### 5.2.3 Higher Level Knowledge Virtualisation

Knowledge Management covers a very large number of concepts. It is not yet clear how this much of this can be covered in CASPAR. There are a large number of projects, for example those connected with the Semantic Web, from which we should be able to benefit.

One of the significant virtualisations which we do need to put in place is that which isolates CASPAR from particular current implementations, and we anticipate developing a number of higher level interfaces which can sit on top of large suites such as RQL\textsuperscript{16} and also the implementations of the SPARQL\textsuperscript{17} standard, illustrated in Figure 16.

---

\textsuperscript{15} \url{http://fits.gsfc.nasa.gov/documents.html}

\textsuperscript{16} \url{http://139.91.183.30:9090/RDF/RQL/}

\textsuperscript{17} \url{http://www.w3.org/TR/rdf-sparql-query/}
5.2.4 Access Control/Trust Virtualisation

As with Knowledge Management there are several approaches and implementations. A virtualisation effort which CASPAR will undertake is to try to identify a relatively simple interface which can be implemented on top of several of these existing systems.

Access Control, Trust and Digital Rights Management are tied together, although they cover, in general, distinct functions and different domains. In brief, Access Control can be distinguished from DRM mainly by the following aspects:

- **Functional**: Access Control focus only on the enforcement of authorization policies, while DRM covers several aspects related to the management of authorization policies.

- **Policy domain**: the Access Control authorization policies lose their semantics and validity once the digital objects leave the information system, while the digital rights have system independent semantics and legal validity.

- **Enforcement extent**: DRM focuses on persistent protection of rights, as it remains in force wherever the content goes, while a digital content that is protected by an information system’s access control mechanism loses its protection once it leaves the system.

Keeping the above characteristics in mind, it can be recognized that both Access Control and Digital Rights Management are needed to govern the Access Administration of OAIS archive holdings. Moreover, both aspects are subjected to changes over time, which need proper attention in order to preserve the access policies that protect the digital holdings.

The interface would have to cover, amongst other things:

- **DRM policy creation**
  - Recognition of rights
  - Assertion of rights
  - Expression of rights

- **DRM policy projection**
  - Dissemination of rights
  - Exposure of rights
  - Enforcement of rights

- **DRM security and cryptography**
- **Access Control technologies**

Access Control policies are defined and are valid within the archival information system.

There may be access restrictions to Content Information that are of different nature: copyright protection, privacy law, as well as further Producer’s instructions. The Producer might wish to allow
access only under the condition that some administrative policies are respected (e.g. defining a group of authorized Consumers, or specifying minimum requirements to be met by enforcement measures). In the long term period, the “maintenance” of all such information within the archive (and between archives) becomes “preservation of administrative information”. In fact, the administrative aspects related to the content access may be subject of some modifications in the long term due to legislative changes, technology evolution, and events that influence the semantics of access policies.

Perhaps a better place to hold administrative information is within the Preservation Description Information (PDI), e.g. as ‘Access Administration’ information, perhaps as part of the Provenance Information, but it may be worth considering adding this as an additional type of PDI.

Access Administration Information wherever placed within PDI, would be the part of PDI that identifies the access restrictions pertaining to the Content Information, in particular to the Data Object, including the legal framework, licensing terms, privacy protection, and agreed Producer's instructions about access control. It contains the access and distribution conditions stated within the Submission Agreement, related to preservation (by the OAIS), dissemination (by the OAIS or the Consumer) and final usage (Designated Community). It would also include the specifications for the application of technological measures for rights enforcement.

5.2.5 Digital Object Storage Virtualisation

The main functions of the Digital Object Storage (“Storage” for short) component has been discussed in section 4.1.2. Storage Virtualisation refers to the process of abstracting logical storage from physical storage. It aims to provide the ability to access data without knowing the details of the storage hardware and access software or its location. This isolation from the particular details facilitates preservation by allowing systems to survive changing hardware and software technologies. Significant work on this has been done in many areas, particularly the various Data Grid\(^{18}\) related projects.

The Warwick Workshop foresaw the need to:

- develop and standardise interfaces to allow “pluggable” storage hardware systems.
- standardise archive storage API i.e. standardised storage virtualisation
- develop language to describe data policy demands and processes, together with associated support systems.
- develop collection oriented description and transfer techniques
- Fuller development of workflow systems and process definition and control.

In more detail, one can, following Moore\(^{19}\), identify a number of areas requiring work to support virtualisation, the most basic being:

- create an infrastructure-independent naming convention
- map administrative attributes onto the logical file name such as the physical location of the file and the name of the file on that particular storage system.
- associate the location of copies (replicas) with the logical name.
- mapping access controls onto the logical name, then when we move the file the access controls do not change.
- map descriptive attributes onto the logical name, and discover files without knowing their name or location.


\(^{19}\) [http://www.dcc.ac.uk/events/warwick_2005/future-warwick.ppt](http://www.dcc.ac.uk/events/warwick_2005/future-warwick.ppt)
Characterization of management policies independently of the implementation needs to cover:

- Validation policies
- Lifetime policies
- Access policies
- Federation policies
- Presentation policies
- Consistency policies

In order to manage ownership of records independently of storage systems one needs details of the

- Data collection
- At each remote storage system, an account ID is created under which the preservation environment stores files
- Management of roles for permitted operations
- Management of authentication of users
- Management of authorization

In order to manage the execution of preservation processes across distributed resources on further needs:

- Management of execution state
- Management of relationships between jobs
- Management of interactions with remote schedulers
6 REPRESENTATION INFORMATION

Representation Information is a key OAIS concept and it is important to examine the concept in more detail. The separation of Structure and Semantics is of special significance. Conceptually this captures the idea that the same information – with the same semantics – can be encoded in different ways. For example a number, for example the number “15”, may have semantics associated with it which tells us that this is 15°C – the average water temperature at a certain position in the Indian Ocean. This number may be encoded in many ways including:

- ASCII: the digits “1” followed by “5” which, in decimal is encoded as “43” followed by “53”, which as a bit sequence is “00110001” followed by “00110101”
- Hexadecimal encoding as the text character “F” – bit sequence is ”1111”
- Big-endian 16-bit integer, with bit sequence “00001111”
- Etc

Another example is an image which can be encoded as a JPEG or as a TIFF, but the semantics of the image is the same in either case.

The distinction between structure and semantics allows us to avoid a combinatorial problem in what if we have N_{sem} sets of Semantics which can be encoded in N_{str} ways then there are N_{sem} * N_{str} possible ways of combining the Semantics and Structure, but in fact one can simply describe the Semantics and Structure separately, providing only N_{sem} + N_{str} descriptions.

6.1 INTRODUCTION

Among the several models within OAIS, the Information Model provides the concepts to support the long-term understandability of the preserved data. This introduces the idea of Representation Information.

![Figure 17 OAIS Information Model](image-url)

The UML diagram in Figure 17 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 interpretation association, the resulting set of objects can be referred to as a Representation Network.

![Figure 18 Representation Information Object](image)

**Figure 18 Representation Information Object**

Figure 18 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.

### 6.2 STRUCTURE

Structure-type Representation Information may be captured in many ways, possibly many ways for the same structure, including:
- An international or community standard describing the format used
- A formal description of the structure – for example an EAST or DRB description.

### 6.3 SEMANTICS

The Semantic content of a piece of information is hard to define. For example does a piece of “raw” (unprocessed) data in some sense have within it the highly processed final results which can be
produced from it? We believe that in most instances the answer will be reasonably clear cut, but it is possible that there will be some grey areas which need some further study. It may also be useful to distinguish between what we might call “Technical” from “Intentional” semantics in a fairly pragmatic way.

6.3.1 Technical Semantics
Technical Semantics are those semantics which can be captured formally in, for example, Data Dictionaries or Ontologies.

6.3.2 Intentional Semantics
Intentional Semantics includes those things which are, at least currently, impossible to capture formally. As an examples one has the meaning which a composer wished to embody in a composition.

6.4 OTHER
The “Other” heading in Representation Information is a very general catch-all term, some examples of which are given in the OAIS reference Model. In particular one has the concept of Software which captures within it Representation Information about the Data Object.

Here we believe that it is useful to introduce some broader concepts which will help when capturing Representation Information.

6.4.1 Behaviour, Actions and Processes
Many Data Objects are bit sequences from which individual pieces of information can be extracted. Many other Data Objects are much more complex – where it is much more difficult to isolate one particular piece of information. Examples include a piece of compiled computer code, or an Oracle database, or a musical patch.

Some information has, as an integral part of its content, an implicit or explicit process associated with it - this could be argued to be a type of semantics, however it is probably sufficiently different to need special classification. Examples of this include embedded queries in databases or other time dependent or reactive systems such as Neural Nets.

The process may be implicitly encoded in the data, for example with the scheme for encoding time dependence in XML data as noted above. Alternatively the process may be held in the Representation Information possibly as software. Amongst many other possibilities under this topic, Software and Software Emulation are among the most interesting (http://www.dlib.org/dlib/october00/granger/10granger.html).

It may be possible to develop a Universal Virtual Computer (UVC) as outlined by Lorie (http://www.rlg.org/preserv/diginews/diginews5-3.html#feature2). However, recognising that one of the prime desirable features of a UVC is that it is well defined and can be implemented on numerous architectures, it may be possible to use something already in place, namely the JAVA Virtual Machine (JVM, http://java.sun.com/docs/books/vmspec/).

6.4.1.1 Rights Enforcement
Some information to make a Data Object finally understandable will be created only after ingestion, either in the preservation life-cycle or at dissemination time.

An example is persistent protection in Digital Rights Management. Technological solutions for rights enforcement are chosen and applied only at dissemination time, using the most up-to-date available solutions. Thus, rights enforcement technology is not going to be preserved along with the Data Object. It is rather enforcement requirements/specifications that must be captured and preserved. The latter information can be considered as PDI.
Then, at dissemination time, specific Representation Information will be packed into the DIP in order to instruct on the tools used to access this handled content. Precise information tied to a specific technology or tool could for instance be stored implicitly in form of software within the DIP’s representation information.

6.4.1.2 Software
In the examples above the Representation Information is embodied in software, which brings with it a large number of dependencies.

6.4.2 Time dependence
Another type of information which might be asked of a Data Object is the state at a particular time in the past. Many, some would say most, datasets change over time and the state at each particular moment in time may be important. This is an important area requiring further research.

Examples include time dependent data such as databases. One way of encoding this is to keep the differentials between versions, as is done in some Version Control systems. In these cases the Representation Information net includes the underlying system of version control.

Efficient storage of a series of snapshots may lead one to store differences or include time tags in the data (see for example Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. On the Propagation of Deletions and Annotations through Views. In Proceedings of 21st ACM Symposium on Principles of Database Systems.). Additional Representation Information would be needed which describes how to get to a particular time's snapshot from the efficiently encoded version.
7 PRESERVATION DESCRIPTION INFORMATION

7.1 INTRODUCTION

Figure 19 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. It is likely that at least some aspects of Provenance will be discipline dependent, as will be Context information.

7.2 AUTHENTICITY

Authenticity is a key concept in digital preservation, and some would argue that is it the pre-eminent concept, in that unless one can prove that the data object is, in some provable sense, what was originally deposited, then one cannot prove that digital preservation has been successful. There are many aspects involved in authenticity. OAIS focuses on what it calls Preservation Description Information (PDI).

A full discussion of the concept of authenticity is beyond the scope of this overview document. However it is worth noting the distinction made by InterPARES, although specifically referring to records, between verification and maintenance of authenticity.

The first is “the act or process of establishing a correspondence between known facts about the record and the various contexts in which it has been created and maintained, and the proposed fact of the record’s authenticity”\(^{20}\). The latter is related to records which “have been presumed or verified authentic in the appraisal process, and have been transferred from the creator to the preserver”.

CASPAR focuses on what InterPARES calls the maintenance of authenticity i.e. providing a continuing chain of evidence about the custodianship and treatment of the information.

A separate position paper is in preparation, excerpts of which may be included in future revisions of this document.

7.3 OAIS PDI COMPONENTS

7.3.1 Fixity Information

OAIS defined Fixity Information as the:

\[^{20}\text{InterPARES Project, Authenticity Task Force, }\textit{Requirements for Assessing and Maintaining the Authenticity of Electronic Records, }\text{March 2002 (http://www.interpares.org).}\]
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 Cyclic 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”. Warming of the vulnerability of any particular type of digest algorithm would be another 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 could be simply a Registry-type entity – 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.

7.3.2 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.

We discuss first name spaces in general and then persistent identifiers in particular.

7.3.2.1 Name Spaces

There are many names spaces in the preservation environment covering, for example, names for files, users, storage systems and management rules. Each of these may change over time as information is handed over in the chain of preservation, or as any single archive evolves. These name spaces, and their associated Access Controls and Representation Information must themselves be managed.
7.3.2.2 Persistent Identifiers

Persistent Identifiers have been the cause of much debate, and there are many proposed systems\(^{21}\). To produce general purpose Persistent Identifiers, which could be used to point to any and all objects, is well known to be challenging, the difficulty being social rather than technological.

A more limited type of Persistent Identifier is the Curation Persistent Identifier (CPID) which was introduced in section 2.3.3 as pointing to Representation Information. It is proposed to focus on the CPID and use that as a base for more general consideration.

It is relatively easy to generate a unique identifier by having a hierarchical namespace,

\[x.y.z\]

each segment or namespace (i.e. each of \(x\), \(y\), \(z\)) forms a hierarchy of naming authorities, and where necessary to generate unique strings some algorithm such as that used by the UUID (http://www.dsps.net/uuid.html) is used. A UUID is a Universal Unique IDentifier which is a 128 bit number which can be assigned to any object and which is guaranteed to be unique. The mechanism used to guarantee uniqueness is through combinations of hardware addresses, time stamps and random seeds.

The difficulty task is to make the link between the identifier (as a character string) to the object to which it points. In particular the bootstrap procedure must be in place, in other words given a string - how does one know what to do with it - where does one start.

The steps involved would be

1. given "\(x.y.z\)" one somehow knows (i.e. the bootstrap step) that one uses some service "\(A\)" with which one can find out what "\(x\)" means i.e. tells one where to go to look up some service ("\(B\)") associated with "\(x\)". "\(A\)" will be refered to here as the bootstrap resolver service
2. using service "\(B\)" we then find out something about "\(y\)" - in particular some service "\(C\)"
3. using service "\(C\)" we then find out something about "\(z\)" - in particular some service "\(D\)" which will point, at last, to the object wanted. This will be referred to here as the terminal resolver service

We presumably can say something about the last service "\(D\)". On the other hand we may have no control over the others in the hierarchy.

Thus we have the issues of - in the specific case of Persistent Identifiers for Representation Information (ReplInfo)

1. the bootstrap into the name resolution system
2. the persistence of each of the name resolvers
3. CASPAR’s own ReplInfo registry/repository

The concept introduced here follows the adage “do not put all one’s eggs in one basket” for step (2). Conceptually one needs to allow multiple name resolution mechanisms in the hope that at least one survives, in order to get to the host (or hosts) which hold the digital object. An XML encoding may look something like:

```xml
<cpid>
  <value>xxxxxxxxxx
    <nameresolver type=n1>http://x.y.z</nameresolver>
    <nameresolver type=n2>DOI:123456</nameresolver>
    <nameresolver type=n3>urn::xx::dd</nameresolver>
  </value>
</cpid>
```

\(^{21}\) See for example the DCC Workshop on Persistent Identifiers available at [http://www.dcc.ac.uk/events/pi-2005/](http://www.dcc.ac.uk/events/pi-2005/), viewed 10 April 2007.
7.3.3 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.

7.3.4 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.

There are a wide variety of approaches to describing, modelling and tracking provenance; a full survey is beyond the scope of this document. Related work includes (amongst many others) CIDOC-CRM, PREMIS and the Chimera Virtual Data Language (VDL). Some projects have focused on formal computer languages for representing the origins and source of scientific and declarative data; VDL falls in this category, as do Semantic Web systems such as W3C’s SPARQL which have explicit fine-grained support for representing the source of pieces of information, and characteristics of that source. Others emphasise an analysis of common concepts (often expressed in some formal ontology language) that capture important aspects relating to Time, Event and Process.

Another consideration is the sharability 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.
8 PACKAGING

OAIS 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.

The OAIS should also avoid holding PDI or Content Information only in the naming conventions of directory or file name structures. These structures are most likely to be used as Packaging Information. Packaging Information is not preserved by Migration. Any information saved in file names or directory structures may be lost when the Packaging Information is altered. The subject of Packaging Information is an important consideration to the Migration of Information within an OAIS to newer media.

The contents of a general Information Package is illustrated in Figure 21

OAIS further introduced a taxonomy of Information Packages, as shown in Figure 22
Figure 22 Information Package Taxonomy

Of these the only one which OAIS describes in detail is the Archival Information Package (AIP), which is conceptually vital for the preservation of a digital object. According to OAIS the AIP is defined to provide a concise way of referring to a set of information that has, in principle, all the qualities needed for permanent, or indefinite, Long Term Preservation of a designated Information Object.

The full AIP is illustrated in Figure 23.

Figure 23 OAIS Archival Information Package (AIP)

There are very many ways of packaging information, both physically as well as logically. CASPAR must provide at least one packaging implementation which can be used in the Testbeds. It should also be possible to provide some level of Virtualisation – possibly related to the “tree” structure of a simple or complex object. In addition there will have to be some aspects of the “on-demand” object, for example where a sub-component in the package has to be uncompressed in order to produce the next level of unpacking which is needed.
9 CONCLUSIONS

The Conceptual Model presented here touches on all aspects of the CASPAR approach to the preservation of digitally encoded information. It will be reflected in the CASPAR architecture. As work progresses in CASPAR it is expected that some improvements and expansion will be needed, and this will be documented in the document Conceptual Model - Phase 2.

---

1 http://public.ccsds.org/publications/archive/650x0b1.pdf