Implementing the PREMIS data dictionary: a survey of approaches

4 June 2007

Deborah Woodyard-Robinson
Woodyard-Robinson Holdings Ltd
For
The PREMIS Maintenance Activity
sponsored by the Library of Congress

ACKNOWLEDGEMENTS

Particular thanks to the following people for contributing their time, expertise and cooperation for this report:

Mathew Black,	National Library of New Zealand
Steve Bordwell,	National Archives of Scotland
Adrian Brown,	National Archives, UK
Priscilla Caplan,	Florida Center for Library Automation, USA
Gerard Clifton,	National Library of Australia
Ruth Duerr,	National Snow and Ice Data Center, USA
Rebecca Guenther,	Library of Congress, USA
Nancy Hoebelheinrich,	Stanford University Libraries, USA
Brian Lavoie,	OCLC, USA
Bronwyn Lee,	Australian Partnership for Sustainable Repositories
Yaniv Levi,	ExLibris, Israel
Justin Littman,	Library of Congress, USA
Julien Masanes,	International Internet Preservation Consortium, France
John Meyer,	Portico, USA
Mark Middleton,	British Library, UK
Gordon Mohr,	International Internet Preservation Consortium, USA
Barbara Sierman,	Koninklijke Bibliotheek, The Netherlands
Susan Thomas,	Oxford University Library Services, UK
Dave Thompson,	Wellcome Trust, UK
Andrew Wilson,	Arts and Humanities Data Service, UK

CONTENTS

LIST OF TABLES

PREFACE

The Preservation Metadata: Implementation Strategies (PREMIS) Working Group developed the *Data Dictionary for Preservation Metadata*, which is a specification containing a set of "core" preservation metadata elements that has broad applicability within the digital preservation community. The PREMIS Data Dictionary (PDD) was released in May 2005 along with a set of XML schemas to support its implementation. Since that time, institutions have begun to implement preservation metadata by providing content for semantic units expressed in the data dictionary or comparing it with planned or existing systems for long-term preservation. Because of the large scale of digital data in digital repositories, it is unlikely that values for semantic units will be supplied by hand, and institutions are looking for guidance as to how to supply these. In addition, the number of semantic units that are specified in PREMIS may seem overwhelming at first glance, and implementation may seem a daunting task.

The Library of Congress, as part of the PREMIS maintenance activity, commissioned Deborah Woodyard-Robinson to provide this study to explore how institutions have implemented the PREMIS semantic units. The goal is to assist the newly established PREMIS Editorial Committee with its first revision of the data dictionary and schemas by understanding the difficulties presented in applying the semantic units and thus improve the specification. In this study sixteen repositories have been surveyed about their interpretation and application of the PDD, with an analysis then made on how the PREMIS core fits with the functions of a preservation repository and which PDD semantic units will be most relevant to certain types of repositories.

Sally H. McCallum
Network Development and MARC Standards Office
Library of Congress
June 2007

1. EXECUTIVE SUMMARY

Long-term digital repositories around the world are looking for guidance on the implementation of preservation metadata and this report examines how the PREMIS Data Dictionary version 1.0 (PDD) is being implemented in that role.

Since publication of version 1.0 of the PDD in May 2005 a number of repositories have been implementing corresponding preservation metadata in new systems or comparing it with planned or existing systems for long-term preservation. Sixteen of these repositories have been surveyed about their interpretation and application of the PDD for this report.

Conformance to the PDD is difficult to measure and open to interpretation.

The **common function use cases** demonstrate how the PREMIS core fits with the functions of a preservation repository. The **common context use cases** can then draw on which PDD semantic units will be most relevant to a certain type of repository.

As observed in the first report of the PREMIS working group, trends such as storing preservation metadata in either XML structures or relational database management systems (RDBMS) and allowing the design of systems to be able to incorporate multiple strategies for digital preservation, continue to hold true.

Very few off-the-shelf tools are being used for implementing preservation metadata. The main three tools in use,

1. DROID/PRONOM (Digital Record Object Identification and format registry),
2. JHOVE (JSTOR/Harvard Object Validation Environment) and
3. the National Library of New Zealand Metadata Extraction Tool,

all relate to technical metadata creation.

Many other implementation methods are being developed in-house for repositories as part of ingest workflow.

Most repositories surveyed identified well with the PDD data model, implementing equivalent metadata entities for intellectual entities, object entities at representation, file and (less commonly) bitstream levels, event entities and agent entities. Rights entities were only occasionally created.

This table is a summary of the research observations broken down by semantic units or groups of semantic units:

Table 1. Summary of this report's research observations in relation to semantic units.

1.1 objectIdentifier	Implemented by all, a standard feature
1.2 preservationLevel	Applied by all repositories but in different manners based on different decision making criteria

1.3 objectCategory	Implemented in XML schema, but implicit in RDBMS
1.4.1 compositionLevel	Mixed uptake of this unit depending largely on type of materials being collected with or without bundling or encryption.
1.4.2 fixity	Most applications are using at least one checksum algorithm
1.4.3 size	A simple function implemented by all repositories
1.4.4 format	All repositories recognise the need for format information. Common tools are often implemented to identify format.
1.4.4.2 formatRegistry	Some implementations include format registry information which may be either an internal or external registry.
1.4.5 significantProperties	Application and definition of significant properties varied widely. Considerable development is still required in this area.
1.4.6 inhibitors	As with compositionLevel, implementation varied on whether a repository collects affected objects.
1.5 creatingApplication	This information can generally be extracted using one of the common tools.
1.6 originalName	Most repositories collect this during the ingest process.
1.8 environment	Rarely recorded in the flat PDD structure. The most functional systems are linking to this information in a separately referenced system to avoid changing object metadata over time as supporting technology changes.
1.9 signatureInformation	Only one repository using signatures and currently implementing the W3C de facto standard.
1.10 relationship	Implementation varies widely. XML based systems record this explicitly and RDBMS tend to record it implicitly.
1.11 linkingEventIdentifier	Known but may not be explicit. Often populated during the event process.
1.12 linkingIntelectualEntityIdentifier	One repository explicitly recorded this. Some link from Intellectual entity not the object.
1.13 linkingPermissionStatementIdentifier	Not currently in use but planned for future use by some repositories.
2 Event entity	Events are usually recorded by most repositories.
2.1 eventIdentifier	Often locally defined or implicit in a system.

2.2 eventType	Thorough controlled vocabularies have been created for some projects. Redundant in systems that define specific event entities rather than a generic event entity.
3. Agent entity	A majority of repositories include some kind of agent entity, often already existing in some other local system.
4. Rights entity	Implementation of a rights entity is not well standardised yet. Often rights may refer to a depositor agreement or statement that applies to a group of objects.

Despite the original aim of the report to find similarities in implementations within common context use cases, there are not yet enough implementations of sufficient maturity to draw conclusions about such typical examples of preservation metadata implementation. In practice the method of implementation, XML vs. RDBMS, has a greater bearing on the implementation methods utilized than the context of the repository.

The PREMIS Data Dictionary semantic units are generally well adopted by the repositories surveyed suggesting there is agreement over the necessity to record the information prescribed although little functionality has been implemented to use the metadata.

Format specific technical metadata requires further development by long-term digital repositories but was deemed out of scope for PREMIS.

Further definition of units such as significant properties and preservation level would be beneficial in revisions of the PDD, as would clearer guidelines on the attainment of PREMIS conformance.

2. INTRODUCTION

The final report of the PREMIS working group[1] including the PREMIS Data Dictionary version 1.0 has been heralded by the international preservation community:

> "The work of the PREMIS Working Group goes a long way towards establishing an international open-source standard for handling meta-data, which will help libraries and institutions around the world to archive digital content"
>
> *- 2005 Digital Preservation Award*[2]

> "The work is intellectually sophisticated, groundbreaking, truly collaborative and international in scope and of great significance for the archival preservation community."
>
> *- 2006 Society of American Archivists Preservation Publication Award*[3]

Since the publication of the PREMIS Data Dictionary version1.0 (PDD) a number of repositories have adopted preservation metadata informed by the data dictionary or have created crosswalks with existing systems. This report aims to assist further development and implementation of the PDD by comparing and discussing the methods used by several of these early implementers.

This report was commissioned by the PREMIS Maintenance Activity, a group tasked to provide ongoing support for the PDD with financial support from the Library of Congress. Originally the aim of the report was to compile guidelines and recommendations for implementation of the PDD in the context of a set of common digital preservation use cases. However, the number of implementations at production stage is still very limited and very few examples of application to a common use case could be found. Where more than one example could be found the application varied so widely that no substantial conclusions could be made.

It is not surprising that it is still taking a long time for repositories to progress preservation metadata implementation beyond planning and development stages into full production. Anyone with experience of trying to implement preservation metadata in a long-term digital repository has quickly discovered the complexities that lie beneath the surface of this contemporary challenge. From the definition of an object to the vagaries of describing the technology requirements, there are still many questions a repository needs to answer specific to their own context. Therefore the results in this document report on the current state of the art of crafting functional preservation metadata and aims to provide guidance using examples that are currently available.

---

[1] *Data Dictionary for Preservation Metadata: Final Report of the PREMIS Working Group* (May 2005) http://www.oclc.org/research/projects/pmwg/premis-final.pdf

[2] http://www.dpconline.org/graphics/advocacy/press/award2005.html

[3] http://www.archivists.org/recognition/dc2006-awards.asp#preservation

Brief history of PREMIS

OCLC and RLG initiated the international working group PREMIS (Preservation Metadata: Implementation Strategies) to define implementable, core preservation metadata, with guidelines and recommendations for its management and use. This work extended the previous activities of the Preservation Metadata Framework Working Group[4], also sponsored by OCLC and RLG.

The complete history and membership of the PREMIS working group has been documented in other places and can be found in detail in the Introduction to the PREMIS Data Dictionary or on the working group web site[5]. The brief history represented here gives attention to the implementation focus of PREMIS.

In order to scope the boundary of the aims of the core preservation metadata being developed by PREMIS the working group defined "preservation metadata" as:

> "*the information a repository uses to support the digital preservation process*."

The group further defined the "digital preservation process" as:

> "*functions to maintain* ***viability, renderability, understandability, authenticity & identity*** *of digital material in a preservation context*."

These functions of the digital preservation process provide the reason why preservation metadata needs to be collected and grounds the group's work in the logical requirements of implementation.

The first report produced by the PREMIS working group in September 2004 was the result of a survey of the cultural environment for creating and using preservation metadata. The report called "Implementing Preservation Repositories for Digital Materials: Current Practice and Emerging Trends in the Cultural Heritage Community" looked at mission, funding, preservation strategy, and access policies with an overall focus on current practice for managing preservation metadata in digital archiving systems.

General trends and conclusions observed in the first report, such as storing preservation metadata in either XML structures or relational databases and allowing the design of systems to be able to incorporate multiple strategies for digital preservation, continue to hold true as will be seen later in this report.

[4] See more information about the Preservation Metadata Framework Working Group at: http://www.oclc.org/research/projects/pmwg/wg1.htm
[5] http://www.oclc.org/research/projects/pmwg/

The work of the group culminated in the release of the "PREMIS Data Dictionary, version 1.0" (PDD) as part of the report entitled "Data Dictionary for Preservation Metadata: Final Report of the PREMIS Working Group" in May 2005.

The PDD was declared to remain a stable document for at least a year to allow the preservation community time to test and discuss the content. Subsequent feedback is expected to contribute to required changes and improvements.

With the dissolution of the working group, support for the PDD was passed to the PREMIS Maintenance Activity organised by the Managing Agency based at the Library of Congress. Support has included creation of an online workspace and international discussion list for a group called the PREMIS Implementers Group (PIG) for self-subscribing members.

METS compatible XML schemas have been developed for implementing the core metadata element set. These schema are maintained in the Network Development and MARC Standards Office of the Library of Congress.

In addition, two further reports have been commissioned by the Managing Agency:

- this report on recommendations for implementing the PREMIS Data Dictionary in the context of a set of common digital preservation use cases,
- and another on rights issues related to the preservation of digital materials, with an emphasis on the metadata requirements needed to document and manage these rights in a digital preservation repository setting, which was released in January 2007[6].

There has been great interest and encouragement for the PDD in the digital preservation community with many projects relying on this comprehensive document to demystify the implementation of preservation metadata.

In August 2006, the PREMIS Managing Agency established an Editorial Committee to resume refinement of the Data Dictionary.

Research and scope

This report drew largely on information provided by repositories listed in the PREMIS Implementation Registry[7] plus a few other high profile projects involving preservation metadata.

The initiatives consulted, and the abbreviations used for them in this report, are listed in Table 2. Several repositories are still in the planning stages of using preservation metadata, a few have started development, some map their existing metadata to the PDD and many are planning further development to their current systems in light of the PDD.

[6] Coyle, Karen (Dec. 2006) Rights in the PREMIS Data Model, http://www.loc.gov/standards/premis/Rights-in-the-PREMIS-Data-Model.pdf
[7] http://www.loc.gov/standards/premis/premis-registry.php

The third column in the table indicates the current stage of development for preservation metadata in each system.

Table 2. Repositories and projects surveyed in this report

Abbrev.	Repository/Project Name	Stage of preservation metadata development
APSR	Australian Partnership for Sustainable Repositories	Planning and documentation
DigiTool	A content management system developed by ExLibris (more a tool than a specific project, included here because it is in the PREMIS Implementation Registry)	Different versions in use in various projects
FDA	Florida Digital Archive, which uses the DAITSS preservation repository application	Mapped to PREMIS, some further development planned
IIPC	International Internet Preservation Consortium	Not an implementation, but a group discussing concepts and tools
KB	Koninklijke Bibliotheek (National Library of the Netherlands)	Mapped to PREMIS, further development in planning
MathArc	A collaborative project of Cornell University Library and Göttingen State and University Library	Project in prototype development
NAS DDA	National Archives of Scotland, Digital Data Archive	Development stage
NDNP LOC	National Digital Newspaper Program, Library of Congress	Production
NLNZ NDHA	National Library of New Zealand, National Digital Heritage Archive	Development stage
NSIDC	National Snow and Ice Data Center, USA	Mapped to PREMIS, further development proposed
Paradigm	A collaborative project of Oxford University Library Services John Rylands Library, University of Manchester	Mapping to PREMIS
Portico	An electronic archiving service	Mapped to PREMIS, some further development planned
SHERPA DP	SHERPA Digital Preservation Project at the Arts and Humanities Data Service, UK	Prototype created, further development continuing
SDR	Stanford Digital Repository	Mapped to PREMIS, further development in planning
TNA	The National Archives, UK	Mapped to PREMIS, further development in planning
Wellcome	The Wellcome Trust, UK	Theoretical

It should be noted that although the repositories and projects outlined in the table above have made commitment to preservation metadata in their systems, the degree of commitment still varies considerably. Some projects have aimed to incorporate the complete PREMIS recommendations while others have selected only a portion that they wish to apply.

The main focus of this report is on the implementation of preservation metadata as outlined in the PDD. It specifically addresses the creation and population of metadata

elements as well as other methods of implementing the PREMIS semantic units, including brief discussion of the tools used for automatic creation of preservation metadata.

The tools discussed in the report are mainly related to metadata derivation and extraction. Other technologies, such as the systems used to implement a digital repository, have not been examined except for small details where it has a direct effect on the preservation metadata.

The surveyed repositories presented at least two distinct approaches to creating systems for preservation metadata management:

1. encoding in an XML schema, often based on METS, or
2. use of a relational database system.

There are also implementations that use a combination of both these methods, for example the FDA uses XML to store the metadata permanently with the object, and redundantly stores the same information in a relational database for ease of use.

The details of how to implement an XML schema or relational database are outside the scope of this report. Discussion on the PIG-list has included a significant amount on the subject of implementing the XML schema for PREMIS entities. More information on this discussion as well as the XML schema for PREMIS[8] can be found on the PREMIS web site.

PREMIS Conformance

Although the PDD gives some general guidelines, the definition of PREMIS conformance is not clearly understood by repositories, or at least not consistently interpreted in implementations (assuming that they were attempting to conform). To be PREMIS conformant increases benefits such as interoperability and assurance that required information for long-term preservation is being captured. Therefore many repositories have an interest in conforming which will affect their implementation of preservation metadata. In practice however it is noticeable how interpretation of the data dictionary varies widely and makes conformance difficult to ascertain.

What does it mean to "conform" to PREMIS?

The PDD states that PREMIS conformance requires a preservation repository to follow the specifications outlined in the Data Dictionary.[9]

That is:

1. Any metadata element sharing the name of a semantic unit in the Data Dictionary will also share the definition of the semantic unit.

[8] http://www.loc.gov/standards/premis/schemas.html
[9] See page 6-1 of the PDD for the discussion on conformance

2. Metadata not defined in the Data Dictionary may certainly be used, but non-PREMIS elements should not conflict or overlap with PREMIS semantic units. i.e. local metadata can be used to extend but not modify the PREMIS semantic units.
3. Data constraints and applicability guidelines in the Data Dictionary must be adhered to.
4. For repeatability and obligation, PREMIS conformance permits more stringent but not more liberal application. That is, a semantic unit defined in the Data Dictionary as repeatable can be treated as not repeatable within a repository, but not vice versa.

Mandatory semantic units represent the minimum amount of information 1) necessary to support the long-term preservation of digital objects, and 2) that must accompany a digital object as it is transferred from the custody of one preservation repository to another.

In general, the mandatory semantic units of the Data Dictionary represent the information that must be able to be associated with any archived digital object in a preservation repository. The specific means of association (e.g., local metadata storage, shared registries, etc.) are implementation issues and outside the scope of the Data Dictionary.

The PREMIS use of the term “mandatory” is different to the general definition of mandatory in other data dictionaries and therefore this is a source for some confusion in implementation. Where normally a “mandatory element” requires that a metadata field must exist and be populated, the PREMIS definition is “A mandatory semantic unit is something that the preservation repository needs to know *independent of how or whether* the repository records it.”[10] A reasonable interpretation of "mandatory" in the PREMIS context is that a value for the semantic unit could be supplied programmatically by the repository as metadata for exchange with other repositories.

Challenges for measuring PREMIS conformance

It is not the place of this report to judge whether the preservation metadata implementations surveyed here are PREMIS conformant, but some observations are given to promote further discussion on conformance issues for future review of the PDD. Issues are discussed in the context of the four points describing conformance listed above.

1. **Any metadata element sharing the name of a semantic unit in the Data Dictionary will also share the definition of the semantic unit.**

Metadata elements with the same name as semantic units are common in repository implementations. The difficulty in measuring against this criteria is where the definition of the semantic unit can be interpreted in different ways.

[10] Page 2-2 of the PDD, italic emphasis in text applied in this document only

An example of this is the broad definition of significantProperties in the PDD, which has resulted in varied interpretation by repositories. A number of repositories appear to have extended their local definition to accept technical properties of formats. These types of properties were specifically stated as out of scope of the PREMIS work and it was recommended that they should be handled by other elements relevant to format-specific technical details.

Occasionally projects have used the PDD definition of a semantic unit but altered the location of that information in the model which can conflict with another definition.

Changing the location of a PREMIS semantic unit is demonstrated in MathArc. They record structural dependencies between web page files as "dependencies" in the environment section. Usage notes for dependencies in the PDD suggest that this type of dependency should be recorded in the "relationships" section as type "structural". The usage notes may not have been interpreted as part of the definition of the semantic unit.

2. **Metadata not defined in the Data Dictionary may certainly be used, but non-PREMIS elements should not conflict or overlap with PREMIS semantic units. i.e. local metadata can be used to extend but not modify the PREMIS semantic units.**

Repositories are encouraged to add detail or granularity in the metadata they collect corresponding to the PDD semantic units. Repositories may add detail such as creating accompanying elements, e.g. SHERPA DP recommend an extra element to accompany preservationLevel to record a reason for the selection of the level. Repositories may include further internal definition to a semantic unit, such as TNA, who provide more elements to define swOtherInformation. Repositories may also implement more complex data models which still incorporate the basic PREMIS concepts. For example TNA prefers to record specific event entities within their system rather than use the generic event entity described in the PDD. This is an acceptable and encouraged variation.

However, the distinction between extending a semantic unit and modifying a semantic unit in metadata is not very clear. Added detail could pose challenges for interoperability, such as a non-repeatable semantic unit originally envisaged as a single value that is expressed in metadata with a complex set of values that don't contradict the definition or purpose of the PDD, but effectively do complicate the content. For example, the KB is considering storing additional information about specific characteristics of a class or collection of objects, such as the context, content, structure, behaviour and appearance. This may be suitable content for the significant properties semantic unit but could provide a more complex content structure than anticipated by the PDD.

3. **Data constraints and applicability guidelines in the Data Dictionary must be adhered to.**

This point insists that data constraints must be met, however the PDD also provides for variations in implementation. It must be possible to be conformant without adhering to a data constraint in the PDD in certain cases. For example, a mandatory semantic unit need not be explicitly recorded as long as the repository "knows" this information and records it in some manner. This is demonstrated by the KB use of compositionLevel. In KB policy they state that in principal they will not archive objects with compression or encryption applied. They do not explicitly record the value "0" as is the data constraint specified in the PDD, but they do know this information and could conceivably export this as metadata containing a default value if required.

Where data constraints require the use of controlled vocabularies PREMIS encourages repositories to develop their own controlled value lists. Examples of controlled vocabularies that differ from the list given in the PDD can be seen in the discussion of the Event Entity semantic unit "eventType" later in this report.

4. **For repeatability and obligation, PREMIS conformance permits more stringent but not more liberal application. That is, a semantic unit defined in the Data Dictionary as repeatable can be treated as not repeatable within a repository, but not vice versa.**

An example of not applying the obligation of a semantic unit as it is stated in the PDD can be seen in practice in some repositories that utilise a relational database structure to link entities. Repositories such as TNA and the NLNZ NDHA do not use explicit event identifiers although within the Event entity eventIdentifier is a mandatory semantic unit. Technically these repositories do "know" how to uniquely identify an event and link the event with the right object, which is the intent of this semantic unit.

The Objective of conformance

It is worth repeating that the aim of PREMIS is to provide guidance on the "core" metadata needed to support digital preservation and to conform to the PDD is desirable to help ensure care for digital objects across time and location.

Naturally conformance to PREMIS does not preclude other metadata elements in an archive. The core of essential information required for long-term preservation will need to be supplemented with other useful information captured in other metadata elements.

3. USE CASES

The original aim of this report was to discuss the implementation of the PDD within the context of a set of common digital preservation use cases.

From the context it was inferred that "use cases" meant the types of function a repository might serve, e.g. library, research data archive, etc. Another definition of "use cases" more commonly used in system engineering is the description of a scenario to convey how a repository would achieve a specific business goal or function.

The term "Use cases" has been examined in both ways. The **common function use cases** demonstrate how the PREMIS core fits with the functions of a preservation repository. The **common context use cases** can then draw on which PDD semantic units will be most relevant to a certain type of repository.

Common Function Use Cases

To understand the functions of a digital preservation repository we need to examine the overall purpose of the system. Naturally the main goal is to preserve digital information.

First, it is helpful to recognize that digital information exists on three integral levels:

- **Physical**, i.e. the storage media holding the binary data
- **Logical** (alternatively called Conceptual), i.e. the encoded representation of the data
- **Intellectual**, i.e. the added context that gives meaning to the data

Digital information will be severely compromised if any one of these aspects is lost.

The functional aims of preservation metadata as stated by the PREMIS working group can be aligned to achieving the preservation of these three levels of digital information as illustrated in table 2.

The OAIS reference model[11], a standard which informs the design of most digital preservation repositories[12], also recognises the physical, logical and intellectual nature of digital information and documents functions to ensure their long-term preservation. The OAIS also goes on to describe the additional information required, i.e. the metadata, to support these functions independent of how these systems could be implemented. These information components form a structure called the "OAIS information model".

[11] http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html

[12] Based on results of the survey documented in the PREMIS report "Implementing Preservation Repositories for Digital Materials: Current Practice and Emerging Trends in the Cultural Heritage Community", September 2004, http://www.oclc.org/research/projects/pmwg/surveyreport.pdf

Table 3. Digital information components in relation to PREMIS functional aims and OAIS metadata

Digital information levels of existence	Corresponding functional aims of PREMIS metadata	OAIS information model components
Physical	Viability	Packaging Information
Logical/conceptual	Renderability	Content Information including Representation Information
Intellectual	Understandability Authenticity Identity	Preservation Description Information including Reference Information Context Information Provenance Information Fixity Information

The PREMIS functional aims and OAIS information model can then be further analysed to discover the relevant, common use cases that would fulfil each level of preservation.

Each use case requires certain types of preservation metadata to support its function. The result of this comparison and analysis can be seen in the table below. The table shows a single correlation between use cases and functions although it may be argued that there is greater crossover than is illustrated. For example, authenticity is often broader than ensuring fixity as represented by the use cases given, it may also require history and provenance information, which also provides context to maintain understandability.

Table 4. Mapping repository functions to functional use cases to core preservation metadata

Function of a digital preservation repository	Relevant use cases	Core preservation metadata required for supporting these use cases
Maintain Viability	- Monitor storage media - Refresh media - Replicate on backup media - Replace media	Content Location Storage Medium
Maintain Renderability	- Monitor technology - Design preservation actions - Perform preservation actions - Supply renderable version	Object Identifier Preservation level Format Inhibitors Environment Relationships – structural
Maintain Understandability	- Record history & provenance - Maintain Context - Understand an object	Creation details Original file name Relationships – context/derivation Rights Events Agents

Maintain Authenticity	- Apply fixity check - Check Fixity - Apply signature - Read signature	Fixity / check-sum details Digital signature details Events
Maintain Identity	- Apply unique identifier - Resolve unique identifier	Object Identifier

This way a direct link between the aims of PREMIS metadata and the functional aims of a repository is illustrated.

This represents a significant concept in the design of preservation metadata for a repository system, which is that metadata must support the functions *in* a system (e.g. monitor technology or record provenance). This in turn means the metadata will also support the function *of* a system (e.g. preserve e-journals).

Common Context Use Cases

It is proposed that the context or environment for application of a repository system will shape the preservation metadata required, e.g. a national library will have different preservation metadata requirements than a commercial research data archive. Note, however, that there were not enough implementations in final production stage to be able to draw enough data to reliably illustrate or draw definitive conclusions about any identifiable context use cases.

Following on with this theory, the mission of a repository will be reflected in the priorities given to specific preservation functions. For example a repository concerned with preserving the "look and feel" of digital objects will place greater emphasis on the detail and accuracy of renderability than a repository that is only interested in basic renderability of content and is happy to normalise all files.

All preservation repositories are essentially required to maintain the viability of storage media to the same degree despite their mission. However the degree of commitment to the other functional aims is flexible. Other possible differences between repository contexts may be, for example,:

- the type of objects they manage and which object categories will need to be represented, i.e. representations, files, bitstreams.
- the characterisation of their user community which will affect their choices surrounding understandability functions, or
- their responsibility for proof of authenticity.

Issues such as these can significantly impact on the preservation metadata of a system according to PREMIS because the PDD asserts that only semantic units that apply to a repository's own situation need to be known by the repository. This means that only applicable object categories need to be managed. There is no obligation to manage

bitstreams, for example, if a repository does not require them. This includes “mandatory” units which are also only required if applicable.

When applying the relationship of functional aims to metadata it is possible to compare the repository context to the functions required, and therefore the metadata required.

Table 5. Example of common context use case effects on preservation metadata requirements.

Common context use case examples	**Characterised by**	**Commitment to functional aim authenticity:**	**Functional use cases for authenticity required**	**Resulting preservation metadata required**
Archives of records containing high evidentiary value	Objects require reliable proof of authenticity	High	- Apply and check fixity - Apply and check digital signatures - Record of provenance	- Fixity / check-sum details - Digital signature details - Event history
Private sector company library	Normalised objects for cost effective preservation planning	Low	- Apply fixity only for internal data processing checks	- Fixity / check-sum details for system functions, not as explicit metadata

In addition, an important distinction in the PDD is that Mandatory means “need to know” rather than “must exist as a metadata element”. PREMIS does not dictate what is required to be explicitly recorded as metadata or whether it may be implicitly stored in database structures or business rules. Repositories should record only what is applicable, e.g. signature information is required only if signatures are used.

4. IMPLEMENTATION OF PREMIS SEMANTIC UNITS

This section steps through the PDD to compare current implementation practices.

As observed in the first report of the PREMIS working group, trends such as storing preservation metadata in either XML structures or relational databases and allowing the design of systems to be able to incorporate multiple strategies for digital preservation, continue to hold true.

The context for a repository can dictate the preservation metadata the repository requires to be recorded, as discussed in the previous section on common context use cases. However, it has been observed that the way the information is recorded and implemented is influenced more by the type of system being developed. The projects and repositories that are focused on using XML structures for storage and transfer mechanisms are creating metadata element based systems, where each piece of metadata is recorded explicitly. However the repositories implementing a relational database management system (RDBMS) record a significant amount of preservation metadata information implicitly within the design of the database structures and business rules. Design of the data model in a RDBMS can implicitly capture information about relationships between entities such as the structure of objects or links to environment information. Business rules applied in a RDBMS can record information that applies to a broad category of objects, whereas this information is captured in a metadata element and repeated for each object in most XML schema based repositories. Some repositories use a combination of both methods.

Most methods for metadata creation, in both types of systems, are developed in-house to be executed as part of an ingest routine. Almost all preservation metadata is created in this early stage of the digital object life cycle.

There are only a few off the shelf tools available to be implemented, such as JHOVE and DROID which are mainly for addressing technical metadata, and therefore many projects seem to be creating systems with the same tools. See Section 4 for more information on Tools.

Data models

Most repositories surveyed identified well with the PREMIS data model[13] although the entities used by each implementation vary. For example, Table 6 below shows equivalent entities used in 4 projects that exist in different contexts.

The main differences with the PDD are that there are very few bitstream object and rights entity equivalents.

[13] See page 1-1 of the PREMIS Final Report

Table 6. Equivalent entities used in repository data models

PREMIS entity	MathArc equivalent	TNA equivalent	Web archive[14] equivalent	NSIDC equivalent
Intellectual Entity		Deliverable Unit (DU)	Collection/Crawl Session	Data set
Object: Representation	Asset/ representation	Manifestation	Site Page	Granule
Object: File	File	File	File	File
Object: Bitstream	-	Bitstream	-	-
Event	Event	Event	Crawl Session	Event metadata
Agent	Agent	Agent authority files	-	Agent metadata
Rights	-	(DU metadata)	-	-

Semantic Units (SU)

Semantic units are the properties of an entity in the PREMIS data model. A semantic unit may be a container for other semantic units or it may be a single unit that relates to a value.

Notation used for Semantic Units discussed below:

SU [SU number] Name of semantic unit			
S: Structural level (containers & units, or unit)	A: Applicability (R=Representation, F=File, B=Bitstream)	O: Obligation (mandatory, optional)	R: Repeatability (repeatable, not repeatable)

The structural level of the semantic units may refer to a single SU (called a unit) or a group of SUs (containers & units) depending on implementation of those units. It is often logical to discuss the application of a whole group of semantic units at once, rather than as individual units. The applicability, obligation and repeatability are taken directly from the PDD.

There are generally two issues to be addressed in the implementation of semantic units. The first is what **values** will be stored to correspond to a semantic unit including the process for deciding which value to use, and the second is how those values are **created** for implementing and recording in the system. For example, for an object identifier the repository must decide the type of identifiers that are required, usually a policy decision, and then implement a tool to generate those identifiers.

[14] This generic web archive reference is built on work from the IIPC. The IIPC is a consortium of web archiving organizations that do not necessarily have consensus on a data model or implementation of metadata.

SU 1.1 objectIdentifier			
S: container & units	A: R, F, B	O: mandatory	R: repeatable

Little is mentioned by the surveyed repositories about internal identifiers because they can be created by most repository systems as a standard feature.

The KB implements National Bibliographic Numbers (NBN) and Portico uses the tool developed by John Kunze called NOID to generate unique archival identifiers.

SU 1.2 preservationLevel			
S: unit	A: R, F	O: mandatory	R: not repeatable

All repositories surveyed account for the preservation level applicable to their objects.

However, repositories do differ in the expression of preservationLevel. Some describe the intention of the repository to provide preservation for a certain type of object (e.g. “isDigitalOriginal”). Others use terms that reflect the current capability for preserving the format of the object (e.g. “bit-level”).

Depending on the purpose of a repository, the choice is usually made to record either a single level of preservation commitment for all the material held within the repository or else provide a small number of options to be selected in relation to one or more properties of an object.

Two types of repository that may choose a single level of preservation were encountered. The first was a repository that has a restricted outlook for preservation due to the type of object they acquire. An example of this is the NSIDC who primarily collect science data sets which they describe as essentially only preservable at a bit or byte level. The data itself does not possess presentation characteristics, for example. Therefore the only decision is whether the data (plus its ancillary documentation and metadata) will be preserved, rather than at what level it will be preserved. The NSIDC however does not yet actually record preservation level and is still evaluating whether there may be some other criteria applicable that they may base a preservation level decision on in future developments.

The second type of repository currently applying only a single level of preservation commitment is an institution that values all of its content equally and aims to apply the same level of commitment regardless of the object. An example of this is TNA where the aim is to apply the same level of preservation commitment to all objects. Another example is the KB, where all material is considered of the highest importance and there is a commitment to retain the “look and feel” of all objects. In both of the above cases where a repository has chosen only one level of preservation commitment they are able to

record the decision at a policy level only and do not require a record of the decision within the metadata of each object.

Interestingly, this decision is earmarked for possible change in future revisions of the KB archiving policies. The KB is considering storing information at a collection level on each of 5 main characteristics of an object. These characteristics, which may be considered the significant properties of an object, are the context, content, structure, behaviour and appearance, and they now believe that each may have its own level of importance in the preservation of any particular class of object and therefore affect the preservation level.

The most common use of preservation level is for a repository to select a value from a locally defined scale of preservation commitment. The table below contains examples of the values used in such scales.

Table 7. Examples of terms used for preservation level

Repository	Preservation commitment		
	High level	**Medium level**	**Low level**
NLNZ	isPreservationMaster	isDigitalOriginal[15]	isAccessCopy
SHERPA DP	00 (Full)	01 (Content-only)	02 (Bit-level)
Portico	Fully supported	Reasonable effort	Byte preserved
FDA	Full (Full preservation)	Bit (Bit-level only)	None (Do not archive)
Deep Blue Michigan	Level 1 (Highest)	Level 2 (Limited)	Level 3 (As-is)

Despite alignment in the table above, the definition and implication of these terms is not directly comparable between repositories. As mentioned earlier, some describe intention and others describe capability, and also the preservation actions applied to an object at one repository in relation to a high level of preservation commitment may not be the same as the actions performed for a high level of preservation commitment at another. E.g. the medium preservation level for FDA is bit-level preservation which is equivalent to the low level preservation commitment for SHERPA DP and Portico.

The criteria that are used to determine the applicable preservation level are also different as can be seen in the table below.

Table 8. Examples of criteria used for determining preservation level

Repository	Criteria used for determining preservation level
NLNZ	Deposit details, object lifecycle
SHERPA DP	File format suitability for preservation, preservation rights
Portico	Format and format validity
FDA	Depositor account agreement
Deep Blue Michigan	Expected longevity of file formats

Ideally the selection of preservation level is made automatically based on a set of criteria that are well defined and encoded in the business rules of the repository. In the interests

[15] An object designated as "isDigitalOriginal" may also be assigned "isPreservationMaster" and provided with the highest level of preservation commitment.

of automation, it may be safest to select a default of the highest level of preservation commitment in a repository unless otherwise stated, a procedure practised by the SHERPA DP.

The source for the criteria used also needs to be reliable, such as parsing a file for validity or depositor account details that are entered by staff. The SHERPA DP notes that depositors should not be asked to assign preservation level directly, because they may easily misinterpret the preservation level and provide misguided information. Depositors may reliably supply straightforward and legitimate criteria for the repository to use for this decision, such as copyright restrictions on the design template for an e-Print that will affect the preservation options available.

Deep Blue Michigan[16] provides a very detailed example of criteria used to set the preservation level in their repository. They apply three levels of preservation support based on the expected longevity of specific file formats. The longevity is determined by evaluating the prevalence of the file format in the marketplace, whether the format is proprietary, the availability of tools for emulation or migration and the availability of local resources to take specific preservation actions.

Most repositories with a scale of commitment choose to apply the preservation level at only one level of object (i.e. representation, file or bitstream) or at the policy level. For example, MathArc chooses to store the preservation level in policy only, NLNZ records the preservation level for representations, while the FDA and Portico assign preservation level at the file level.

An interesting addition to the record of preservation level is observed by the SHERPA DP. They require a second metadata field to note the reason that preservation level was selected. This will enable better understanding of the decision in the future. This "reason" metadata element is not considered core metadata and is only required to be stored locally with a repository and does not necessarily have to be made available more widely.

While all repositories implement preservationLevel and conform to the current definition it is easy to see that values only have local significance are not interoperable between repositories.

SU 1.3 objectCategory			
S: unit	**A:** R, F, B	**O:** mandatory	**R:** not repeatable

The object category states whether a set of metadata applies to a representation, file or bitstream. Two distinctly different methods are used to handle this semantic unit

[16] Deep Blue Michigan provides significant detail on their use of preservation level but was not surveyed for any other PDD elements for this report. The repository appears to be using a system that does not yet support full preservation metadata. http://deepblue.lib.umich.edu/about/deepbluepreservation.jsp

depending on the repository system implemented. It is either used as a metadata element or as an implicit structural feature of the repository system.

Where XML schema, such as the PREMIS:object XML schema, are used to store metadata the semantic unit is implemented as a metadata element populated using a controlled vocabulary and used explicitly to link the metadata to a particular object category. For example MathArc uses “representation” and “file” within an XML construct to differentiate between the two object categories it manages. The SDR intends to use a controlled vocabulary containing all three categories, “representation”, “file” and “bitstream”.

The second method of implementation is where this semantic unit is not explicitly recorded but implied by the structure of the repository system, such as in a relational database. Data models will commonly relate object entities in a hierarchy that reflects the object category. Representation entities will tend to link to one or more file entities and file entities may link to zero or more bitstream entities. Therefore this hierarchy implicitly records the object category by the placement of the entity/object in the hierarchy. The NLNZ system for example manages representations and files in this manner. The NLNZ NDHA will also manage bitstreams via metadata.

TNA uses the structure of the repository system to record this information and employs the concept of a Deliverable Unit (DU), which describes the conceptual record. This equates to the PREMIS Intellectual Entity. A DU can have multiple Manifestations, which equate to PREMIS Representations. Each Manifestation can comprise multiple files, and they have adopted the PREMIS distinction between bitstreams and filestreams, to describe the component parts of a file.

The IIPC consists of many organisations involved in web archiving, each with a different approach. They are working together to develop standards and tools for web archiving. Although the IIPC is not implementing the PDD, their work on metadata relates to object categories such as “site”, “page” and “file”, where both sites and pages refer to an equivalent for the PREMIS representation level object and files are PREMIS file level objects. It can be useful to differentiate between these object categories within a single PREMIS category for managing the objects in an appropriate manner.

Data sets may be described as “granules” equivalent to representation level and “files” at file level, as is often used by the NSIDC.

A repository that only manages objects on one level may not record this semantic unit in either way as it will be the same for all objects. For example the NDNP applies PREMIS descriptions to all objects at the file level and does not explicitly record the object category.

SU 1.4.1 compositionLevel			
S: unit	**A:** F, B	**O:** mandatory	**R:** not repeatable

Most repositories appear to intend to record the composition level of objects whether they plan to collect bundled or encrypted objects or not.

A repository, such as the KB , may be reluctant to store objects with compression or encryption and therefore, rather than record this individually with objects, they record this as a business rule to satisfy the mandatory status of this SU. (Mandatory meaning the archive must "know this information") A repository with a policy not to store compressed or encrypted objects but wanting to validate their XML schemas, such as MathArc, may record this mandatory element with a default value of "0".

The KB may reconsider their decision not to store compressed data if in the future they decide to use the WARC format for web archiving (which may be used with or without compression) which will then require recording a new composition level to enumerate the unbundling required.

Others such as SDR and NAS DDA do not intend to hold many compressed or encrypted objects but will provide the facility to record composition level directly as described in the PDD in case the situation arises.

TNA and NLNZ NDHA know the value of the composition level in their systems implicitly through relationships between bitstreams and/or files and bitstreams. Each compressed or encrypted file or bitstream has a format recorded which will indicate the encryption or compression format of that bitstream, and that will have a relationship to another bitstream of the next level with its own format information, etc.

SU 1.4.2 fixity			
S: container & units	**A:** F, B	**O:** optional	**R:** repeatable

Most applications are using at least one checksum algorithm to produce a message digest, the most popular are MD5 and SHA-1. NLNZ NDHA and FDA use both MD5 and SHA-1 checksums. Just one repository surveyed, Portico, reported using SHA-512.

All projects applied the checksum at the file level only. None claimed to be using a checksum at bitstream level.

Checksums are usually calculated during the ingest workflow. Or a checksum that was created before it was received at the repository will be checked during the ingest process. For example at the FDA, any file checksums provided on ingest are verified and the object is rejected if there is a mismatch.

Where a repository has committed to one type of checksum algorithm they may or may not be recording the message digest algorithm in a metadata element, it may simply be a business rule.

TNA has created a second element, Fixity Method to supplement the messageDigestAlgorithm known to them as Fixity Type. Therefore Fixity Type describes the type of algorithm used (e.g. "MD5 digest algorithm") while Fixity Method describes the tool used to produce the messageDigest (e.g. "MD5 Summer 1.1.0.22"). A controlled vocabulary is used for Fixity Type.

The message digest semantic unit is commonly used in the surveyed repositories and the PDD definition is adhered to.

The message digest originator is recorded by only half of the repositories surveyed. To enable automation, where it was used, it was either a default value, added automatically in workflow or chosen from a controlled list of users who may have initiated the process.

SU 1.4.3 size			
S: unit	**A:** F, B	**O:** optional	**R:** not repeatable

All repositories record the size of files and are using bytes as the unit of measurement. The KB also records the size of the representation as a whole although this is considered not applicable at this level in the PDD.

Capturing the file size is a common system function, generally added to an ingest workflow. For example the NAS DDA proposes to use a Visual Basic function to populate this element.

SU 1.4.4 format			
S: container	**A:** F, B	**O:** mandatory	**R:** not repeatable

PREMIS requires either the formatDesignation or formatRegistry semantic units to be recorded. Some repositories have developed complex format identification routines and others use only the simplest and possibly less accurate methods.

APSR recommends using both formatDesignation and formatRegistry in case the registry fails or is unavailable when needed, and also suggests it will be useful information to store locally for reporting and management functions.

Despite the PDD statement that “format designations in common use, such as MIME types and file type extensions, are not granular enough … without the addition of version information” a number of repositories are using MIME types without adding any version

information. Only one of these repositories provides an additional but optional field for format registry information. This is most likely for ease of automated capture but falls short of PREMIS recommended practice.

Portico uses MIME types in the identification process but also adds more format information unless more accurate identification is not possible.

Standard practice for Portico is to use JHOVE in combination with the standard Unix utility called BSD File to identify file formats, but is considering the use of DROID. Using two different tools assists with the identification of files that are bad or mislabelled. The initial step of the process is to attempt verification of the file based on MIME type using JHOVE. If this process fails then BSD File is used to attempt identification. If the type of file identified is different than the type it was originally expected to be it may have been a mislabelled file and can then attempt to be verified against the new format type.

FDA first look up the file extension in a list of known extensions for supported file types. If there is a match, an attempt is made to identify the "magic number" to verify the file is what it says it is, and if successful the file is parsed to obtain the exact version. If no match is made (i.e. it is an unsupported format) they use the UNIX utility ffident to provisionally identify the format. Such a file is considered an "unknown" type but the provisional format identification is retained.

TNA uses DROID to identify the file format and version using a combination of internal and external signatures and assigns a PRONOM Unique Identifier (PUID) to be stored, which is equivalent to the format registry key. The PUID acts as a pointer to detailed format and environment information in PRONOM.

NLNZ NDHA uses the NLNZ Metadata Extraction Tool to identify the format and version of the most common files in their collections. The primary format for a file will be associated with the file level metadata. If additional bitstream metadata can be extracted, then it will be inserted in to a second metadata record specifically for the bitstream information.

SU 1.4.4.2 formatRegistry			
S: container	**A:** F, B	**O:** optional	**R:** repeatable

A mix of local and external format registries are used by the surveyed repositories.

The KB is using their own locally developed internal format registry, however it is not linked from the object metadata but exists in their Preservation Manager system. See SU 1.8 Environment for more detail.

APSR will prefer to use universally available and comprehensive registries and will allow the ability to provide links to more than one registry.

Portico has been developed with the intention of linking to the Global Digital Format Registry (GDFR)[17]. The GDFR has recently received further funding but is not yet operational.

As previously stated, TNA uses DROID to provide a PUID as a format registry key to link to the PRONOM registry. This is incorporated in the system design so values do not need to be recorded for the format registry name or role.

NAS DDA intends to use PRONOM as its registry and the format registry fields will be populated by DROID.

See more information on PRONOM and DROID under the section on Tools on page 44.

Format registries work especially well for objects of a format that can be found in more than one place or application. However it is possible that a repository archiving science and research datasets could be dealing with data sets almost entirely with unique formats created specifically for each project. These repositories are less likely to be able to use a general format registry due to the disparate nature of its objects. These data sets are often highly proprietary depending on the specific instruments that have created them. For example, the datasets received by the NSIDC from the Moderate Resolution Imaging Spectroradiometer (MODIS) flying aboard both the AQUA and TERRA satellites are completely different from the data sets associated with the Cold Land Processes Field Experiment, such as the CLPx ISA Snow Pit Measurements data set. These datasets require widely different supporting documentation and it is essential to record specific format information for each data set.

SU 1.4.5 significantProperties			
S: unit	**A:** R, F, B	**O:** optional	**R:** repeatable

Projects encountered did not yet have a vocabulary fully developed for this SU and they also vary in the level at which they apply significant properties, from files to collections of intellectual entities.

NDNP use significant properties to record any rules that a file has been permitted to violate. That is, NDNP have developed detailed profiles for all of their file formats, and in some cases, it is appropriate to allow those profiles to be violated.

NLNZ NDHA records significant properties in relation to the representation level in their preservation policy. It is unlikely that this will be recorded individually for objects.

[17] http://hul.harvard.edu/gdfr/

The KB does not expect to record significant properties at an object level. They relate significant properties to a class of objects or "collections".

Significant properties in SHERPA have been structured to apply specifically to e-prints. The significant properties of an e-print identify the particular properties that must be maintained through subsequent preservation action (e.g. migration) or may have some influence upon the preservation action. The most common examples of significant properties for an e-print include the intellectual content (text and images), as well as the layout of the document. Additional properties that may be recorded in SHERPA are likely to be file format-specific technical characteristics and therefore are outside the scope of the PDD and this report.

TNA describe two conceptually different types of property that are significant to digital objects. They discuss a mixture of invariant properties of an intellectual entity and the technical properties of a representation. Invariant properties are those properties of the record which are significant to its authenticity that must be preserved over time, and across different manifestations. TNA associates these properties at the intellectual entity level because they relate to the conceptual record. TNA plans to measure invariant properties in any given representation by analysis of the component files. This measurement will allow validation of the results of migration and provide for the definition of allowable tolerances for these properties.

Technical properties change with each manifestation of a representation. These properties recorded by TNA are format specific object characteristics which are not in scope for PREMIS or this report.

Therefore we see that the invariant properties described by TNA are the equivalents to the PDD significant properties apart from their application to the intellectual entity which is a level that the PDD does not address.

Invariant properties are related to record types and are modelled as name/value pairs in the TNA system, which allows easy extensibility. These templates of properties can then be linked to particular intellectual entities but not at any level lower than that. The TNA believes that changes to representations through processes such as migration, provide no certainty of a one to one correspondence between files in two representations, so persistent properties cannot be associated with them and therefore must apply to the intellectual entity.

Stanford uses significant properties for technical metadata applicable to all formats, but not included in some format technical metadata schemas. Again, these technical properties are not significant properties as defined in the PDD.

NDNP, SHERPA, TNA and Stanford all attribute some technical properties of format specific object characteristics to significant properties. Perhaps this illustrates a misunderstanding of what PREMIS intended significant properties to address, but it is

certainly an indication of the need for further work on format specific preservation metadata.

SU 1.4.6 inhibitors			
S: container & units	**A:** F, B	**O:** optional	**R:** repeatable

A number of repositories do not record inhibitors. For example, the KB will not allow objects to be submitted if they contain inhibitors as a matter of archive policy and therefore will have none to be recorded.

TNA will record inhibitors as significant properties of objects rather than a separate set of metadata elements specifically for inhibitors. The FDA records inhibitors discovered during the ingest process, but stores the information as a result of the format validation event rather than a property of the object. FDA does not record inhibitor target or inhibitor key.

The AHDS SHERPA Project does expect to receive a small number of e-prints with inhibitors, mainly methods intended to restrict access. They prefer to have an inhibitor free version of the object as their preservation master but suggest maintaining the inhibitor as if it is a significant property when migrating or transforming an object. Their proposed controlled vocabulary for inhibitor types list specific types of encryption or password protection and closely mirrors the list given in the PDD:

- DES encryption
- PGP encryption
- Blowfish encryption
- 128-bit RC4 Password protection
- Certificate protection

NLNZ NDHA are considering applying details about access inhibitors at the representation level. This information can be collected in several different ways: automatically populated using metadata extract tools; manually or automatically assigned by ingest utilities used by staff; or manually assigned by the donor.

SU 1.5 creatingApplication			
S: container & units	**A:** R, F, B	**O:** optional	**R:** repeatable

Creating application information is stored in certain types of file headers and can be easily extracted by tools such as JHOVE or the NLNZ Metadata Extraction Tool for a majority of standard objects.

Direct mapping will depend on the information available in a file and could depend on the version of creating application. For example, it may be common for a DateTime tag to

exist in file headers corresponding to the creatingApplicationDate, however the format of that information may not be ISO standard as required by PREMIS, or the application name and version may be concatenated into one tag such as “Software” in a TIFF header.

NLNZ NDHA intend to automatically populate creating application using metadata extract tools for a majority of objects in their archive. Options will also exist for creating application metadata to be manually or automatically assigned by ingest utilities used by staff or manually assigned by the depositor.

Creating application will also be populated by most systems when an event such as migration creates a new version of the digital object. FDA also records creating application during local processes for creation of files, like creating localized and normalized versions.

SDR expect the data provider depositing the object to include this information in as detailed a manner as possible.

NSIDC also requires extensive information about creation of the data to be supplied by the depositor, though in most cases the data does not come a standard application such as MS Word, Acrobat, etc. , rather it is often the result of a custom software algorithm developed by the science community. Therefore, while they typically do manage to capture information like when a particular file was created, the thing of importance to the science community is to capture the description of the algorithm used to process the data thereby creating the data set. This is what allows the results of research to be replicated, which is key to the scientific process and central to the integrity of the data (and any conclusions drawn from research using the data). This information can only be supplied by the creator/depositor.

SU 1.6 originalName			
S: unit	**A:** F	**O:** optional	**R:** not repeatable

Most repositories collect the original file name during the ingest process. It is a function that is straightforward to automate with standard file management functions. Only two repositories surveyed do not store the original name in a metadata element.

MathArc states that the original name must be available in all three information packages, the SIP, DIP and AIP, but internally the files are referenced by a unique identifier. When disseminated the files may be renamed with the original name.

The KB does record the original name also but due to the batch method of supply from a depositor they note it is often a meaningless batch identifier.

SU 1.7 storage			
S: container & units	**A:** F, B	**O:** mandatory	**R:** repeatable

While all repositories currently know how to locate their objects, few are recording the values explicitly in metadata.

The NLNZ NDHA assigns a location value for files, a process which is managed by the archive system. Where bitstream information can be extracted, they also intend for the metadata extractor or format identification tool to be able to record the file offset and bitstream length for locating bitstreams. The content location type and storage medium are considered implicit in the system and are not explicitly recorded in object metadata.

The FDA records the content location value for files and the content location type and value for bitstreams. These values are created by ingest processes. The storage medium is known by the system and referred to by the current system name "TSM" (Tivoli Storage Manager) from which the tape unit could be inferred.

The KB can infer whether an object is stored on optical or tape storage depending on the function of the object (i.e. preservation or access)

Repositories such as the TNA are aware of storage details however the majority of the information used to manage the storage is provided within the scope of the storage management system and therefore not explicitly recorded in the object metadata.

SU 1.8 environment			
S: container & units	**A:** R, F, B	**O:** optional	**R:** repeatable

Environment information is rarely recorded in the flat structure prescribed in the PDD, and when it is there appears to be little or no functionality attached to the metadata. However, there are a few more complex solutions in development.

Two notable projects, the KB and TNA, are developing systems for handling this information. This has grown from the complex nature and constant changes of environment components to support the use of digital objects. Both systems relate the format information for the object to technical requirements including software and hardware.

The KB relates the format information of their objects at representation level to environment information stored in their "Preservation Manager" system. The Preservation Manager registers information on the file formats stored in their repository, using a structure consisting of "Preservation Layer Models" (PLM) and "View Paths". The PLM describes how the format is related to software and hardware that runs on different conceptual layers in a system. The layers represent similar concepts to the

PREMIS semantic units for software hardware and dependencies. The data format is the top layer followed by separate layers for each component of software application, operating system and reference platform required. Description of each layer includes attributes such as “name”, “version” and “patches”.

A “View Path” is an instance of the preservation layer model related to a file format. It is preferable that more than one view path exists for each file format. That means there are multiple ways for a format to be accessed which will increase its chances of longevity.

A particular benefit of the view paths is that a change may occur in the technology available and this can be reflected in the creation or deprecation of a view path without having to update object metadata.

A stand-alone trial version of the Preservation Manager was successfully tested in 2004 and a production version is intended to be released in 2007. This version will be integrated with the DIAS system for their e-Depot to provide direct links to their environment information.

Similarly, TNA is developing the PRONOM registry to provide environment information related to formats. The technical environment required to support access to objects is described in a very similar way to the PDD environment information but it is not stored directly in object metadata. The format of an object is described with a PUID (PRONOM Unique Identifier) which points to a detailed description in PRONOM. Within PRONOM the format information will provide software information required for use of the object. In turn a detailed record for the software can also be found in the PRONOM database which includes PREMIS semantic unit equivalents plus more as listed in Table 9. Required operating systems and hardware components required are also listed.

Table 9. PREMIS Software semantic units and equivalent metadata fields in PRONOM

PREMIS data dictionary semantic unit	Equivalent PRONOM field/fields
swName	Name
swVersion	Version
swType	Category
swOtherInformation	Description; Other names; Identifiers; Operating systems; Default file format; Language; Supported until
swDependency	Technical dependencies, Related software

TNA are also considering modelling any additional technical requirements or dependencies for a particular file where these are additional to the generic requirements of the format as listed in PRONOM.

Similar to the KB Preservation Manager, PRONOM is a separate system from the repository and it is only linked to object metadata. However, TNA does plan to generate a plain English summary of the required environment to store in the metadata for each representation to support on-line access by users.

Initially, the PUID scheme applied only to the format in which a digital object is encoded. Formats were considered a particular priority for such a scheme, as no existing, universally applicable system of identifiers provides for this. The scheme has been adopted as a recommended encoding scheme for describing file formats in the latest version of the UK e-Government Metadata Standard[18]. The scheme is designed to be extensible, and is currently being expanded to include other classes of representation information in PRONOM, such as software, hardware, compression algorithms, and character encoding schemes.

PUIDs can be expressed as Uniform Resource Identifiers using the info:pronom/ namespace, details of which are available from the info URI registry. The National Archives is developing a range of services to expose PRONOM registry content for remote querying and retrieval. The next release of PRONOM will include a PUID resolution service, together with SOAP and REST interfaces. There are also investigations into the possibility of exposing PRONOM as a metadata repository to support the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH).

Currently the links from the TNA digital archive to the information in PRONOM are created manually but TNA intends to extend PUIDs to apply to all technical environment components (e.g. software, hardware), which will allow the definition of any technical environment as a simple set of PUIDs. This will be similar to the approach by the KB's View Paths in the Preservation Manager, but a significant difference is that PRONOM is intended to be a publicly available resource that can be networked into other systems.

SU 1.9 signatureInformation			
S: container & units	**A:** F, B	**O:** optional	**R:** repeatable

Only one repository surveyed stated that they are using digital signatures, the NDNP at LOC, and they implemented these before the PDD was complete and adopted the W3C's *XML-Signature Syntax and Processing (XML Signatures)* de facto standard for encoding digital signatures to add the metadata to their METS records. The standard is more detailed than the PREMIS signature semantic units. For comparison of PDD signature information with the de facto standard see the PDD page 4-8.

SU 1.10 relationship			
S: container & units	**A:** R, F, B	**O:** optional	**R:** repeatable

Relationships are always a complex issue in a repository and this is reflected by the variety of ways they have been implemented in the repositories surveyed.

[18] See page 32, "2.12 Format", e-Government Metadata Standard Version 3.1, 29 August 2006, http://www.govtalk.gov.uk/documents/eGMS%20version%203_1.pdf

PREMIS describes two types of relationships in this section, “structural” for relationships between components, and “derivation” relationships to express provenance. Projects that are using the relationship sub-types suggested by PREMIS have adopted only a subset of the values listed in the PDD. The subset chosen is different in each project.

Four of the repositories surveyed do not currently record any specific relationship information, although the storage structures they use may group objects with structural relationships together in an information package.

Structural relationships between files may also be recorded in XML using the METS structMap section, which is an implementation decision recommended by the APSR project for exchange of objects between repositories via METS.

MathArc also uses the structMap section in METS for the expression of some, but not all, relationships. The relationship semantic units are only used to store derivation relationships for provenance information. Relationship type is always “Derivation” and the subtype is always “has predecessor” as backward links only are supplied when the creating/migrating event occurs. This derivation association must also have a linked event although the event identifier need only be local to the METS document which will contain the event information. The object and event sequences provided are always “0”.

Interestingly MathArc also discusses relationships between files where they may link to each other as in a web page which has internal links to other files. These relationships are treated as technical dependencies which are described in the Environment dependencies section instead.

FDA relationships point one way towards a file from either the representation or the bitstream. Representations relate to files with a structural “has part” relationship, and bitstreams use structural “is part of” relationships. Therefore sibling relationships between files can be inferred from these two.

In the relational database system at TNA the structural relationships are implicit in the data model design and sub-types are not required. Related object identifier values only are recorded explicitly. Event sequence will also be implicit, relying on dates and database structure to maintain these relationships.

Identifier types are often local or standard in the repositories and therefore are not explicitly recorded.

None of the projects specify how they intend to populate these elements except the derivation relationships that are described as part of an event which are related automatically as part of the process.

SU 1.11 linkingEventIdentifier			
S: container & units	**A:** R, F, B	**O:** optional	**R:** repeatable

It is common practice for most implementations that event identifiers are local to the system and therefore are of a known local identifier type which does not need to be recorded explicitly for each object.

There is a mix across projects of implicitly or explicitly recording the event identifier value and they appear to be used at either file level only or representation and file depending on how the events are perceived to relate to objects. No projects explicitly associate events with bitstreams using elements equivalent to these semantic units.

None of the repositories state how the elements are populated but it is commonly inferred that it is part of the design of the event process to record these details.

SU 1.12 linkingIntellectualEntityIdentifier			
S: container & units	**A:** R, F, B	**O:** optional	**R:** repeatable

Only the FDA uses this semantic unit explicitly. The repository selects the type of identifier from a controlled vocabulary and the identifier value is populated from metadata supplied by the depositor.

The NLNZ NDHA implies this value through database structure.

TNA expects the link to be explicit from the intellectual entity and not the objects.

Several other implementations do not intend using these semantic units.

SU 1.13 linkingPermissionStatementIdentifier			
S: container & units	**A:** R, F, B	**O:** optional	**R:** repeatable

None of the repositories surveyed have implemented this yet, but two have stated they may. NAS DDA and APSR have proposed to use a linking identifier here. NAS DDA will link Representations to Permission records in anticipation that rights will apply equally to all Objects within a Representation.

Often the permission or rights agreements are applicable to a group of objects and not explicitly recorded with the individual object. See the Rights entity discussion for more information.

TNA do record a link, but from the equivalent of their intellectual entity, not the representation.

SU 2 Event entity		
S: container & units	**O:** mandatory	**R:** repeatable

Events are usually recorded by repositories, but the type of events and actual implementation may vary considerably.

APSR regard ingest and events that change the object to be the most important to be recorded in a repository.

NAS DDA proposes to implement the full event entity. They expect to automatically populate all the elements within an event process.

For the FDA, only local events which are performed by their DAITSS preservation repository application are recorded.

The NLNZ NDHA uses an "audit trail" events entity for events such as virus checking, validation and digital provenance actions. The entity includes only the event date, detail, outcome and agent.

TNA prefer to record specific events within their system rather than use a generic event entity for all events. These are explicitly modelled in each case, although they all tend to follow the basic model of event date/time, event agents, event process and event outcome. The TNA system also has a generic event entity, which can be used to capture any other event in a simple log. TNA believes that this design is a trade-off between flexibility and usability, because event types that need to be queriable are much better modelled explicitly.

SU 2.1 eventIdentifier		
S: container & units	**O:** mandatory	**R:** not repeatable

MathArc, Stanford and the FDA explicitly record an event identifier, and APSR recommends using them.

In MathArc and FDA the eventIdentifier consists of an identifier type and a value. FDA records a local constant for the value of the identifier ("FDA") as well as the value. In MathArc, both type and value are used according the project partner's preservation system. The identifier is only required to be unique within each METS stream. This MathArc event identifier is used to link from a relationship to an event.

Portico and TNA use a different structure than is used by the PREMIS model and they do not require an explicit identifier for the event entity.

SU 2.2 eventType		
S: unit	**O:** mandatory	**R:** not repeatable

It is interesting to compare the implementation of event type in a number of projects with a well developed use of a generic event entity to see which events of the PDD suggested starter list are in use and what controlled vocabularies are being adopted.

Table 10. Comparison of controlled vocabularies in use for eventType.

PREMIS	Portico	MathArc	SHERPA DP	FDA
Capture			Capture	
Compression				
Deaccession			Deaccession	
Decompression				
Decryption				
Deletion		Deletion	Deletion	DEL (deleted file)
Digital signature validation				
Dissemination				D (disseminated)
Fixity check	EventChecksum-Verified		Fixity check	VC (verified checksum)
Ingestion			Ingestion	I (ingested)
Message digest calculation	EventChecksum-Computed		Message digest calculation	
Migration		Migration	Migration	M (migrated to)
Normalization	Event-TransformedFile			N (normalized to)
Replication	EventDataCopied			RM (refreshed media)
Validation	EventFormat-Verified EventFormat-VerFailed		Validation	
Virus check	EventVirus-Scanned		Viruscheck	CV (checked for virus)
	EventFormat-Identified			
	EventTmdExtracted			
	EventStatusInactive			
	EventPreservation-LevelChanged			CPD/CPU (changed preservation level downward /upward)
	EventFileAdded			
	EventFileCreated			
	EventFormatChanged			
		Replacement		
		UpdateAsset-Metadata		
		Inconsistency-Discovered		
			Resub_request	

PREMIS	Portico	MathArc	SHERPA DP	FDA
				DLK (down-loaded link)
				L (localized to)
				WA (withdrawn by archive (e.g. a superseded version))
				WO (withdrawn by request of owner)
				Unknown

MathArc is particularly concerned with transferring assets within a distributed archive, therefore the events detailed in their metadata are of particular functional use and will cause repository actions when received by a partner.

SHERPA DP has redefined a subset of the PREMIS event list to be locally relevant. For example:
Migration is defined as:

> "A preservation strategy in which a transformation creates a version of a digital object in a different format, where the new format is compatible with contemporary software and hardware environments. This may be applied to an event that involves the creation of an AIP or DIP. For Sherpa DP, the normalisation event is incorporated into the migration event, to reduce complexity and avoid confusion."

Validation is defined as:

> "The process of comparing an e-print with a standard and noting compliance or exceptions. The event is likely to be performed on ingest into the preservation repository."

Plus they have added other events to their list such as "Resub_request", which is defined as:

> "The AHDS Preservation Service has requested the institutional repository resubmit an e-print. This may apply when an e-print made available for transfer to the AHDS Preservation Service is found to be corrupt. (this is not a PREMIS event type)"

SHERPA DP also states that actions performed during the normal maintenance of the institutional repository or preservation archive, such as the creation of off-site backups, should not be recorded.

Portico and the FDA have also added a number of event types that are not covered by the PDD list. Generally additional event types cover adding and removing archived objects, and adding or editing metadata.

A number of other projects have listed the same suggested events as in the PDD but they are likely to develop further as they come closer to actual implementation.

The type of event is not recorded as a metadata element at TNA, however they do have several event types that are explicitly modelled.

TNA records these event types: Custodial History, Preservation History (pre-ingest), Selection, Accession, Transfer, Virus Check, Access Review, Identification, Validation, Property Extraction, Transformation, Redaction, Fixity Check.

NLNZ NDHA records event details for virus checking, validation, and a more general digital provenance action. This last event type however is covered by creatingApplication in the PDD.

Migration in some form is the one event type that is listed in all implementations. Also popular to record as an event are virus-checking, fixity checking and deletion.

SU 3 Agent entity		
S: containers & units	**O:** optional	**R:** repeatable

A majority of repositories use some form of Agent entity. However, their implementation of agents appear to differ significantly. This is often due to other systems in place in an organisation that also contain entities for agents such as people and organisations. This was expected by PREMIS and is one reason why there is little detail in the agent entity in the PDD.

TNA, NLNZ NDHA and FDA all handle agents in other parts of their system but suggest this could be mapped to the PREMIS agent entity.

Portico, MathArc, APSR and NAS DDA appear to use dedicated agent entities for use with preservation metadata and not necessarily another part of the system. APSR and NAS DDA mention including software in the list of possible agents types. They suggest using agents for a person, organisation or software as is listed in the PDD.

SU 4 Rights entity		
S: containers & units	**O:** optional	**R:** repeatable

Rights are yet another complex area that is handled very differently across repositories and may or may not correspond to the Rights entity in the PDD.

NAS DDA propose to use all the metadata elements corresponding to the rights semantic units and the entity will be linked to the representation level.

Portico currently has a placeholder for Rights Metadata, and this only links the metadata record to the depositors contract that is associated with the content. The contracts or agreements are also stored in the archive.

NLNZ NDHA will generate a permission statement based on information manually entered in a form by the depositor and repository business rules (i.e. library policies). This will generally be an automated process at the repository end.

TNA records intellectual property rights and describe access conditions relating to records, closure, disturbing content etc. at the intellectual object level (i.e. not representation, file or bitstream). This information simply identifies the copyright holder and any restrictions.

MathArc are using OAI collection based rights only.

Another variation on applying rights is to link the rights to a category of objects rather than individual objects such as the practice at the SDR. The rights entity therefore does not necessarily link to individual objects, however all the PDD elements are adopted to record rights information except for the grantingAgent. SDR populate the rights metadata using a template that provides some of the values and controlled vocabularies.

Rights for preservation do not concern some repositories as much as recording the responsibility for preservation. NSIDC, for example, does not use an equivalent to the PDD rights entity. "With scientific data, if you have the data at all, you generally have the right if not the technical capacity to make as many copies of the data as you need to ensure preservation."[19] Research data repositories may hold data sets so large there could be technical or financial problems with even holding a back-up copy of the data. NSIDC would find it useful however to record preservation responsibilities. Where NSIDC is the primary archive, it is NSIDC's responsibility to ensure the preservation of the data, whether that be by holding back-up copies of the data or negotiating to have other organizations provide either a back-up copy of the data or the ability to regenerate the data on request. In other cases, where NSIDC is not the primary archive, NSIDC's responsibilities for the data are different. Explicit information on responsibilities and organizational agreements would assist decision making processes in these situations.

[19] "A New Approach to Preservation Metadata for Scientific Data, A Real World Example" by R. Duerr, R. Weaver and M.A. Parsons, NSIDC (pre-print 2006 IEEE IGARSS Conference)

5. TOOLS

Metadata generation and extraction

There are three main tools available for capturing preservation metadata, JHOVE, DROID and the NLNZ Metadata Extraction Tool, and they all relate to technical metadata. Some simple tools that can provide file size or original file name can be found in common programming libraries. Most metadata that requires application of local business rules still needs to be developed in-house.

The table below from the APSR Presta Report summarises the capability of the three main tools and a more detailed discussion of how they can satisfy PREMIS semantic unit requirements follows.

Table 11. Summary of functions covered by tools from APSR Presta report[20]

Tool	Identify format (Tentative)	Identify format (Confirm)	Identify versions	Validate format	Collect generic file MD	Collect material type MD	Collect file format MD
DROID	Yes [546 formats]	Yes [159 formats]	Yes	No	No	No	No
NLNZ-MET	Yes [15 formats]	(Some)	(Some)	No	Yes	Yes	Yes
JHOVE	Yes [52 formats]	Yes [52 formats]	Yes	Yes	Yes	Yes	Yes

Note that metadata extraction can only retrieve from a file what is already there or can be derived from what is there. For example, only if the file type hosts information about the creating application can that information be extracted, and the quality of the extracted information can only be as good as the quality of information provided by the creating application to populate that tag.

A few other less common but potentially useful tools for preservation metadata generation are also discussed in this section.

DROID Tool and PRONOM Registry

The National Archives of England, Wales and the United Kingdom (TNA) has developed a valuable technical registry called PRONOM and an accompanying tool for file identification called DROID. The registry was created to provide key details and

[20] "PREMIS Requirement Statement Project Report" Bronwyn Lee, Gerard Clifton and Somaya Langley, National Library of Australia July 2006 http://www.apsr.edu.au/publications/presta.pdf

objective information about the file formats, software products and other technical components required to support long-term access to digital objects and is now available online.

DROID (Digital Record Object Identification) is a software tool developed for use with PRONOM to perform automated batch identification of file formats. DROID uses internal and external signatures to identify and report the specific file format versions of digital files. These signatures are stored in an XML signature file, generated from information recorded in PRONOM. New and updated signatures are regularly added to PRONOM, and DROID can be configured to automatically download updated signature files from the PRONOM website via web services.

DROID is a platform-independent Java tool. It provides a documented, public application programming interface (API), for ease of integration with other systems, and can be invoked from a Java graphical user interface (GUI) or command line interface. The results can be configured for output in XML, CSV or printer friendly formats.

Although DROID was initially limited to file identification, the next version of PRONOM will extend the use of PUIDS to software, hardware, compression algorithms, and character encoding schemes. Both DROID and PRONOM are constantly being developed and enhanced to provide further functionality.

TNA have now developed a characterisation framework which, after running DROID, automatically queries PRONOM for available validation and property extraction tools, then automatically deploys them for the relevant objects. The architecture allows any third party tool to be used, via a Java wrapper. The first such tool to be incorporated is JHOVE.

DROID aims to identify file formats as conclusively as possible and attempts to assign a format name and where possible a version and a PRONOM Unique Identifier (PUID). Therefore currently, when a positive identification is made the DROID output could be configured to populate metadata elements corresponding to PREMIS semantic units:

- objectCharacteristics / format / formatDesignation / **formatName**
- objectCharacteristics / format / formatDesignation / **formatVersion**
- objectCharacteristics / format / formatRegistry / **formatRegistryKey**

Therefore a further two elements could be automatically assigned or recorded as business rules for the system if no other format registries are expected to be used:

- objectCharacteristics / format / formatRegistry / **formatRegistryName**
- objectCharacteristics / format / formatRegistry / **formatRegistryRole**

However when a “tentative” identification is made another solution to recording the format may need to be implemented to stay conformant with PREMIS. A DROID “tentative” identification will often produce a list of possible format matches, versions and PUIDs, but a list of file formats cannot be accommodated by PREMIS because the format designation group of semantic units is specified in the PDD as non-repeatable.

Future enhancements of PRONOM aim to enable automated online access to format and environment information for repository systems. This would allow access to environment information, risk assessment and preservation services applicable for a file format via use of the PUID for that format. See the PRONOM web site for progress and news of other enhancements: http://www.nationalarchives.gov.uk/pronom/

The PRONOM registry provides a searchable web database of technical information about file formats, the software tools required to access them, and the technical environments required to access them. Users can search for formats and software using a variety of criteria, such as format or software name and file extension. PRONOM also holds information about support periods for software products and can be queried on this basis. In addition to on-screen viewing, registry information can be exported in XML, CSV and printer-friendly formats. The PRONOM website allows users to submit new information for inclusion in PRONOM.

Below is an example of some of the details in a format record in the PRONOM Registry. There are also links to supporting documentation and external and internal signatures provided in the registry that are not shown here.

Table 12. Sample of information in a PRONOM Registry format record.

Name	Hypertext Markup Language
Version	4.01
Other names	HTML (4.01)
Identifiers	PUID: fmt/100 MIME: text/html Apple Uniform Type Identifier: public.html
Family	
Classification	Text (Mark-up)
Disclosure	Full
Description	The Hyper Text Markup Language (HTML) is a mark-up language designed for the rendering of information via a web browser. It was originally defined as a highly simplified subset of SGML, but is now an international standard, and is maintained by the World Wide Web Consortium (W3C). A HTML document consists of nested elements, each of which may have attributes and content. It begins with an HTML Document Type declaration, defining the HTML version and Document Type Definition (DTD) to which it conforms. HTML 4.01 contains minor editorial revisions to the HTML 4.0 specification.
Orientation	Text
Byte order	

Related file formats	Has lower priority than Extensible Markup Language (1.0) Has priority over Hypertext Markup Language Is subsequent version of Hypertext Markup Language (4.0)
Released	24 Dec 1999
Supported until	
Developed by	World Wide Web Consortium
Supported by	None.

NLNZ Metadata extraction tool

National Library of New Zealand Metadata Extraction Tool Version 1.0

The metadata extraction tool uses a combination of Java and XML to filter and process extracted metadata. The modular design of the tool ensures that it is extensible and scalable. Separate adapters for each file type can be built to allow incremental development of the tool with each adapter processing only the information contained in the files of that file type. If a file type is unknown the tool applies a generic adapter which extracts data that the host system 'knows' about any given file such as size, filename, and date created, until an adapter is written for the new file type

The tool can handle complex dependencies within and between objects, and will process objects with varying levels of complexity ranging from single, simple objects such as a TIFF file to multi-file web sites. Adapters are currently available for MS Word 2, MS Word 6, Word Perfect, Open Office, MS Works, MS Excel, MS PowerPoint, TIFF, JPEG, WAV, MP3, HTML, PDF,GIF, and BMP.

While the tool is designed to support the Library's digital preservation work, the XML output can be configured to support other business processes and business requirements, for example the extraction of metadata for resource discovery.

The extract tool has both a Microsoft Windows interface and a UNIX command line interface. This enables work to be automated through batch processing or processed on an individual basis as required. In either mode the application opens files as read-only, ensuring no inadvertent write operation can be carried out, thus maintaining the integrity of the original files. As the tool only reads header information the extraction process is fast.

Values that can be extracted from the formats that the tool has adapters for include format name and version as well as:

- objectCharacteristics / inhibitors / **inhibitorType**
- objectCharacteristics / inhibitors / **inhibitorTarget**
- objectCharacteristics / inhibitors / **inhibitorKey**

- creatingApplication / **creatingApplicationName**
- creatingApplication / **creatingApplicationVersion**
- creatingApplication / **dateCreatedByApplication**

Download National Library of New Zealand Metadata Extraction Tool Version 1.0 and its documentation from:
http://www.natlib.govt.nz/en/whatsnew/4initiatives.html#extraction

JHOVE

The JSTOR/Harvard Object Validation Environment (JHOVE) is an open source, extensible framework developed to provide format-specific identification, validation, and characterization of digital objects.

JHOVE is made available under the GNU Lesser General Public License (LGPL).

JHOVE uses an extensible plug-in architecture. Any metadata tags embedded in a file can be extracted using an appropriate JHOVE module. The modules to handle various formats can be customized and output from JHOVE is controlled by flexible output handlers. The initial release of JHOVE includes modules for arbitrary byte streams, ASCII and UTF-8 encoded text, TIFF, HTML, XML, JPEG, JPEG2000, PDF, AIFF and WAVE audio; and text and XML output handlers. Development of extra modules for different formats is encouraged. For example Portico has created a module for SGML.

Most technical metadata fields can be populated by JHOVE, and some event metadata may also be performed and captured. JHOVE must have compatible modules for the file types being parsed to be of maximum utility.

Given a recognized format containing appropriate metadata, it is possible for JHOVE to capture these elements:

- objectCharacteristics / **size**
- objectCharacteristics / format / formatDesignation / **formatName**
- objectCharacteristics / format / formatDesignation / **formatVersion**
- objectCharacteristics / inhibitors / **inhibitorType**
- objectCharacteristics / inhibitors / **inhibitorTarget**
- objectCharacteristics / inhibitors / **inhibitorKey**
- creatingApplication / **creatingApplicationName**
- creatingApplication / **creatingApplicationVersion**
- creatingApplication / **dateCreatedByApplication**
- **eventDetail**

Download JHOVE from: http://hul.harvard.edu/jhove/distribution.html
Access further documentation at: http://hul.harvard.edu/jhove/

GDFR

Global Digital Format Registry

This project based at Harvard University Library and OCLC, recently funded by the Andrew W. Mellon Foundation for two years, intends to provide a distributed network of representation information for digital file formats. Any preservation institution will be able to participate, both contributing and using the resources. Considerable development is still under way but the result should be able to provide information about format name, version, registry details and other technical, format specific metadata.

For more information see: https://collaborate.oclc.org/wiki/gdfr/about.html

Xena

(XML Electronic Normalizing of Archives)

Xena is an open source tool developed at the National Archives of Australia designed to recognize and automatically transform files into open document formats. The first part of the process that Xena uses incorporates an extensive file recognition routine, however the tool currently available as open source performs the entire transformation process. Within the internal workflows at the NAA it is possible to perform the recognition of files only and stop. The public release does not yet include this capability, but possible future developments may incorporate this feature.

Download Xena from: http://xena.sourceforge.net/
More information on Xena from: http://xena.sourceforge.net/index.html

NOID

http://search.cpan.org/dist/Noid/noid

Projects such as Portico use the NOID utility in the generation of their unique identifiers. NOID creates generators that can produce persistent, globally unique identifiers for any objects being handled in a repository. The identifier generators, called “minters”, efficiently create, track and bind unique identifiers also called “noids” (nice opaque identifiers) which can be used within naming schemes such as ARK, PURL, URN and DOI.

The noids produced are durable for the long term as they are strings that carry no widely recognisable meaning. This semantic opaqueness helps persistence because the identifiers are free of language specific meanings that may change over time and between collections if objects are managed in a different manner in future.

The minters created are formed by NOID based on a template using BerkeleyDB as the underlying database. The template includes settings for the form, number and intended longevity of the noids to be created by the minter - NOID also includes the ability to choose the term (long-term, medium-term or short-term) that noids will be required to last for. The noids produced can be generated in either random or sequential order within the designated namespace. If required, a check character can also be produced within the noids that can be used to discover transcription errors.

NOID is available to download under a BSD-type open source license from:
http://search.cpan.org/~jak/Noid/
An example of using noids in production of an ARK (Archival Resource Key) can be found at: http://www.cdlib.org/inside/diglib/ark/

Digital Asset Management Systems and PREMIS

Systems such as DSpace and Fedora currently have very limited capability for supporting the PDD at present. This is identified as a possible weakness and future developments are likely to consider possible benefits of addressing PREMIS.

Commercial products such as DigiTool from ExLibris and DIAS from IBM are gaining incentive to support the PDD. DigiTool is listed in the PREMIS Implementation Registry and therefore it is discussed in some more detail below. More information on DIAS can be sourced directly via the KB, Project Kopal and IBM.

DigiTool

DigiTool from ExLibris can store all metadata elements corresponding to the Event, Rights and Object entities in the PDD. (Agents are not currently implemented). The object and its metadata in an XML container within DigiTool

Metadata corresponding to PREMIS semantic units is added/edited manually (except events and identifiers). An NBN generator for object identifiers has been applied to the installation of DigiTool for one of their customers and DigiTool captures local event details.

Some elements in the system have different names to the PREMIS semantic units:
The equivalent of the event entity is “history metadata”, the object entity is called “preservation metadata”, and the rights entity is “PREMIS Rights” (there are different rights metadata for different purposes).

DigiTool assigns the PREMIS definition to the elements in their system and strongly recommend their customers adhere to the recommendations.

6. CONCLUSIONS

There are not yet enough implementations of sufficient maturity to draw conclusions about typical examples of preservation metadata implementation within common context use cases. All areas, except perhaps research data archives (such as NSIDC), appear to be approaching the implementation of preservation metadata in similar ways. While the metadata are similar, the implementation in research data archives may differ in scale, data management practices and heterogeneity of data.

Automation of the extraction of preservation metadata by common tools is limited and only addresses technical metadata. JHOVE, NLNZ metadata extraction tool and DROID/PRONOM are the most widely used externally available tools for preservation metadata creation and extraction of technical metadata.

Some automatic creation and population of other metadata elements is being developed in-house as part of repository systems, particularly within the ingest and workflow software. There are currently no widely available tools to be adopted for this purpose.

As recognized in the PREMIS survey results published in September 2004, two types of implementation models continue to dominate:

- XML schema used to store metadata (e.g.) follows semantic unit descriptions more closely.
- Relational database system used to store metadata and implement preservation functionality where many PREMIS elements become implicit in design of the data model used. (e.g. TNA, NLNZ)

Most of the work thus far in these organizations tends to focus around the creation, capture and management of the preservation metadata and little appears to have been done on the ability to provide functionality using the metadata as it has been recorded.

Most technical semantic units such as identifiers, size and format are addressed in reasonably standard ways. A notable exception is the implementation of environment semantic units. There is a trend emerging among a number of repository systems to record environment via a separate system involving a type of format registry to manage changes and updates.

Semantic units that deal with determining business rules, policy and individual interpretation have provided a much broader implementation variations. For example significant properties and preservation level are both open to more local interpretation.

A reasonably common variation in the interpretation and application of significant properties is the inclusion of technical metadata. NDNP, SHERPA, TNA and Stanford all attribute some significant properties to technical properties of format specific object characteristics. PREMIS specifically ruled technical metadata specific to format types as out of scope for the data dictionary. Perhaps this illustrates a misunderstanding of what

PREMIS intended significant properties to address, but it is certainly an indication of the need for further work on format specific preservation metadata.

Significant properties are broadly recognised as important details to record but there is much work still to be done on their practical identification, measurement and recording in repository systems.

The definition of preservation level in the PDD has resulted in some variations of implementation. It is not necessarily clear in the PDD whether the semantic unit describes the intended level of preservation support as described as it is 'expected to be applied' in the definition or whether it should be the current level of preservation capability as stated by the '"preservability" of the format' in the rationale of the semantic unit. Repositories have reported using this semantic unit in varying senses, and the ambiguity may result in confusion among repositories or between repositories and depositors.

Revisions of the PREMIS Data Dictionary are likely to include further specification of these two semantic units.

Overall a more detailed statement on the motivation, definition and measurement of PREMIS conformance would be helpful in progressing the implementation of preservation metadata.

7. BIBLIOGRAPHY

PREMIS Working Group (2005) *Data Dictionary for Preservation Metadata: Final Report of the PREMIS Working Group*, May 2005
http://www.oclc.org/research/projects/pmwg/premis-final.pdf

PREMIS Working Group (2004) *Implementing Preservation Repositories for Digital Materials: Current Practice and Emerging Trends in the Cultural Heritage Community*, September 2004
http://www.oclc.org/research/projects/pmwg/surveyreport.pdf

Lee, B., Clifton, G. and Langley, S. (2006) *PREMIS Requirement Statement Project Report*, July 2006, National Library of Australia
http://www.apsr.edu.au/publications/presta.pdf

Brandt, O., Enders, M., Kehoe, B. and Rosenkrantz, M. (2005) *MathArc metadata schema for exchanging AIPs.* version 1.3, 3 November 2005
http://www.library.cornell.edu/dlit/MathArc/web/resources/MathArc_metadataschema_v1.3.doc

MathArc (2005) *Math Arc Scenarios, a work in progress*, Version 2005-09-28
http://www.library.cornell.edu/dlit/MathArc/web/resources/MathArcScenarios.doc

Caplan, P. (2006) FCLA Digital Archive Data Dictionary
http://www.fcla.edu/digitalArchive/pdfs/Archive_data_dictionary.pdf

Florida Center for Library Automation (2006) *DAITSS Overview*
http://www.fcla.edu/digitalArchive/pdfs/DAITSS.pdf

Florida Center for Library Automation (2006) *DAITSS and PREMIS Conformance*, version 1.0, March 2006
http://www.fcla.edu/digitalArchive/pdfs/PREMISConformance.pdf

Koninklijke Bibliotheek (2006) *The Preservation Manager for the e-Depot*, web site
http://www.kb.nl/hrd/dd/dd_onderzoek/preservation_subsystem-en.html

Van Wijk, Caroline (2006), *KB and Migration: Working Document* Version: 0.2, National Library of the Netherlands (Koninklijke Bibliotheek) 7 August 2006. Retrieved 4 January 2007 from:
http://www.kb.nl/hrd/dd/dd_projecten/KB%20and%20Migration.pdf

Kunze, J. and Russell, M. () *noid - nice opaque identifier generator commands*, documentation and download web site
http://search.cpan.org/dist/Noid/noid

Kunze, J. (2003) *Archival Resource Key (ARK)*, California Digital Library, USA
http://www.cdlib.org/inside/diglib/ark/

Center for International Earth Science Information Network (CIESIN) (2005) *Data Model for Managing and Preserving Geospatial Electronic Records* Version 1.00, June 2005, Columbia University, New York, USA
http://www.ciesin.columbia.edu/ger/DataModelV1_20050620.pdf

Masanes, J. (2005) IIPC Web Archiving Metadata Set
http://www.iwaw.net/05/masanes2.pdf

Masanes, J. (2005) Metadata for Web Archiving
http://www.dcc.ac.uk/events/fpw-2006/fpw_2006_JMmetadata.pdf

Verhoeven, H. (2006) SIP Interface Specification v 2.5 IBM Nederland
http://kopal.langzeitarchivierung.de/downloads/kopal_DIAS_SIP_Interface_Specification.pdf

Project kopal (2006) *koLibRI: kopal Library for Retrieval and Ingest: Documentation* Die Deutsche Bibliothek / Staats- und Universitätsbibliothek Göttingen, Germany
http://kopal.langzeitarchivierung.de/kolibri/koLibRI_V0_5_beta_Documentation.pdf

Steinke, T. (2006) Universal Object Format: An archiving and exchange format for digital objects. Project kopal. Frankfurt, Germany
http://kopal.langzeitarchivierung.de/downloads/kopal_Universal_Object_Format.pdf

Steinke, T., translation by Wollschläger, Dr T. (2005) *LMER: Long-term preservation Metadata for Electronic Resources* Version: 1.2, Project kopal. Frankfurt, Germany
URN: urn:nbn:de:1111- 2005051906
http://www.ddb.de/standards/pdf/lmer12_e.pdf

Parsons, M. A. and Duerr, R. (2005) "Designating User Communities for Scientific Data: Challenges and Solutions" *Data Science Journal*, Volume 4, 24 August 2005 pp31-38

Fetterer, F. and Smolyar, I. (2005) "On the Creation of Environmental Data Sets for the Arctic Region". *NSF 05-39, Arctic Research in the United States*, Volume 19, Spring/Summer 2005
http://www.nsf.gov/pubs/2005/nsf0539/nsf0539_14.pdf

Duerr R., Parsons, M.A., Marquis, M., Dichtl, R. & Mullins, T. (2004) "Challenges in long-term data stewardship". *Proc. 21st IEEE Conference on Mass Storage Systems and Technologies.* NASA/CP-2004-212750, pp.47-67. College Park, MD, USA.

Hunolt, G. (1999) Global Change Science Requirements for Long-Term Archiving: Report of the Workshop, Oct 28-30, 1998, National Center for Atmospheric Research, Boulder, Colorado, USA

R. Duerr, R. Weaver, M.A. Parsons (2006) "A New Approach to Preservation Metadata for Scientific Data: A Real World Example" *Proc. 2006 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2006)*, Vol 1, pp305-308, Denver, Colorado, USA

Owens , E. (2006) *Automated Workflow for the Ingest and Preservation of Electronic Journals*, Portico; Princeton, New Jersey, USA
http://www.portico.org/news/Archiving2006-Owens.pdf

Fenton, E. (2006) "Building to Preserve: An Overview of Portico", PowerPoint presentation *Digital Preservation Management Workshop*, May 2006, Cornell University, USA
http://www.portico.org/news/DigitalPreservationWorkshop.051806.FINAL.pdf

Owens, E. et al (2005) "A Format-Registry-Based Automated Workflow for the Ingest and Preservation of Electronic Journals", PowerPoint presentation, *Digital Library Federation Fall Forum 2005*, Charlottesville, Virginia
http://www.portico.org/news/Archiving2006-Owens.pdf

Stanford Digital Repository (2005) Stanford Digital Repository – Format Scoring Matrix
http://dlib.org/dlib/december05/johnson/Table1.pdf

Anderson, R., Frost, H., Hoebelheinrich, N. and Johnson, K. (2005) "The AIHT at Stanford University: Automated Preservation Assessment of Heterogeneous Digital Collections" *D-Lib Magazine* December 2005 Volume 11 Number 12
http://www.dlib.org/dlib/december05/johnson/12johnson.html

Cabinet Office, e-Government Unit, Technical Policy Team, Metadata Policy Co-ordinator, (2006) *e-Government Metadata Standard Version 3.1*
http://purl.oclc.org/NET/e-GMS_v3_1

Wheatley, P. (2004) Institutional Repositories in the context of Digital Preservation *DPC Technology Watch Series* Report 04-02, March 2004
http://www.dpconline.org/docs/DPCTWf4word.pdf

National Information Standards Organization and AIIM International (2002) Data Dictionary—Technical Metadata for Digital Still Images NISO Z39.87-2002
AIIM 20-2002
http://www.niso.org/standards/resources/Z39_87_trial_use.pdf

Brown, A. (2006) Digital Preservation Technical Paper 1: Automatic Format Identification Using PRONOM and DROID, Issue: 2, 7 March 2006, The National Archives, London, UK
http://droid.sourceforge.net/wiki/images/b/b4/Technical_Paper_1_-_Automatic_Format_Identification_v2.pdf

Brown, A. (2006) Digital Preservation Technical Paper 2: The PRONOM Unique Identifier Scheme Issue: 2, 27 July 2006, The National Archives, London, UK http://www.nationalarchives.gov.uk/aboutapps/pronom/pdf/pronom_unique_identifier_scheme.pdf

(2004) *Global Digital Format Registry (GDFR) Data Model v.4* Rev. 2004-01-12 http://hul.harvard.edu/gdfr/documents/DataModel-v4-2004-01-12.doc

(2006) *JHOVE2: A Next-Generation Architecture for Format-Aware Digital Object Preservation Processing*, Rev. 2006-06-01 http://hul.harvard.edu/jhove/JHOVE2-proposal.doc

Sytec Resources (2004) *NLNZ Metadata Extraction Tool: User Guide*, Version: 2.0 http://www.natlib.govt.nz/files/Preservation/docs.zip