PREMIS PREservation Metadata Implementation Strategies

PREMIS Tutorial:
Understanding & Implementing the PREMIS Data Dictionary for Preservation Metadata

PREMIS Tutorial
iPRES 2010 Vienna, Austria
September 19, 2010

PRESENTERS

- Priscilla Caplan, Florida Center for Library Automation
- pcaplan@ufl.edu

- Angela Dappert, British Library
- Angela.Dappert@bl.uk

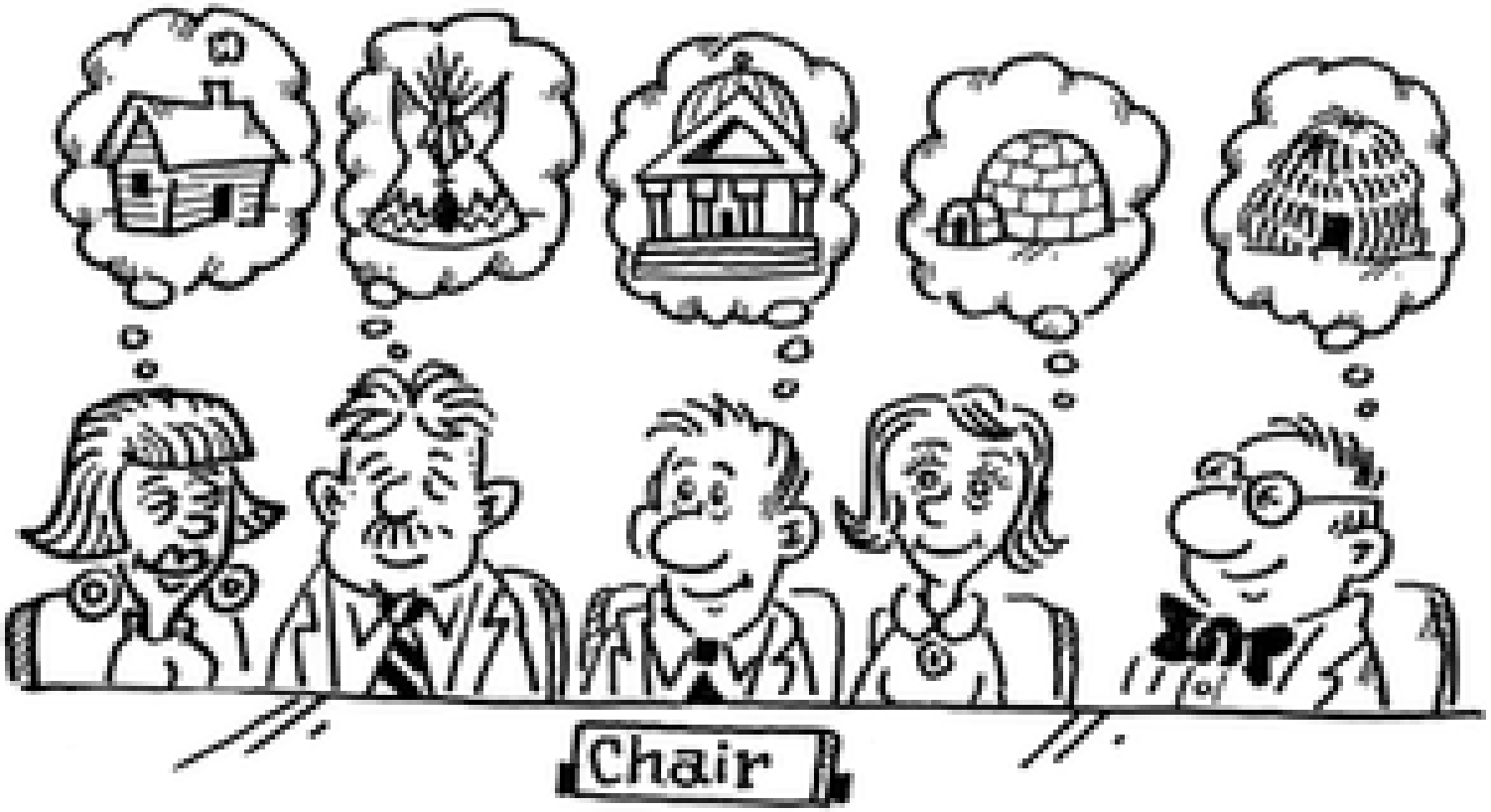
- Markus Enders, British Library
- Markus.enders@bl.uk

- Karin Bredenberg, National Archives of Sweden
- Karin.Bredenberg@riksarkivet.ra.se

GOALS

- Background and context of PREMIS Data Dictionary
- Discuss PREMIS data model, identifiers, and relationships
- Discuss semantic units defined in the Dictionary
- Discuss major implementation issues
- Show ways of representing PREMIS in XML (METS)
  - PREMIS in METS toolkit
- Discuss institutional experiences in working with the PREMIS Data Dictionary

INTRODUCTION: BACKGROUND AND CONTEXT



Some background …

- **Pre-2002:** various preservation metadata element sets released
  - Different scopes, purposes, underlying models/assumptions
  - No international standard; little consolidation of expertise/best practice

- **June 2002:** Preservation Metadata Framework
  - International working group (jointly sponsored by OCLC, RLG)
  - Comprehensive, high-level description of types of information constituting preservation metadata based on OAIS

- **Post-2002:** Need implementable preservation metadata, with guidelines for application and use, relevant to a wide range of digital preservation systems and contexts
  - Motivated formation of **PREMIS Working Group**

PREMIS Working Group

- June 2003: OCLC, RLG sponsored new international working group:
  - ***PREMIS: Pre****servation* ***M****etadata:* ***I****mplementation* ***S****trategies*
- Membership:
  - **> 30 experts from 5 countries, representing libraries, museums, archives, government agencies, and the private sector**
  - Co-Chairs: Priscilla Caplan (FCLA), Rebecca Guenther (LC)
- Objective 1: Identify and evaluate alternative strategies for encoding, storing, managing, and exchanging preservation metadata
  - PREMIS Survey Report (September 2004)
- Objective 2: Define implementable, core preservation metadata, with guidelines/recommendations for management and use

PREMIS Data Dictionary

- **May 2005:** *Data Dictionary for Preservation Metadata: Final Report of the PREMIS Working Group*
- **March 2008:** *PREMIS Data Dictionary for Preservation Metadata, version 2.0*
- **March 2010:** Japanese translation of *PREMIS Data Dictionary* version 2.0
- Includes PREMIS Data Dictionary, context/assumptions, data model, usage examples
- XML schema to support implementation

http://www.loc.gov/standards/premis/v2/premis-2-0.pdf

Some guiding principles …

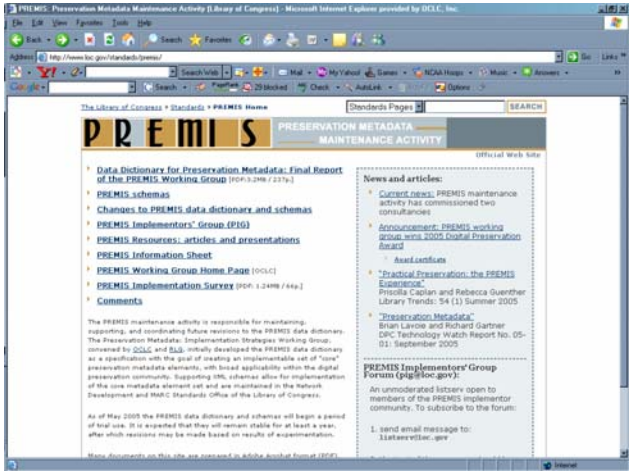
- "Implementable, core, preservation metadata":
  - **Preservation metadata**: maintain viability, renderability, understandability, authenticity, identity in a preservation context
  - **Core**: What most preservation repositories need to know to preserve digital materials over the long-term
  - **Implementable**: rigorously defined; supported by usage guidelines/recommendations; emphasis on automated workflows
- "Technical neutrality":
  - Digital archiving system: no assumptions about specific archiving technology, system/DB architectures, preservation strategy
  - Metadata management: no assumptions about whether metadata is stored locally or in external registry; recorded explicitly or known implicitly; instantiated in one metadata element or multiple elements
  - Promotes flexibility, applicability in wide range of contexts

Scope

- What PREMIS DD **is**:
  - Common data model for organizing/thinking about preservation metadata
  - Guidance for local implementations
  - Standard for exchanging information packages between repositories

- What PREMIS DD **is not**:
  - Out-of-the-box solution: need to instantiate as metadata elements in repository system
  - All needed metadata: excludes business rules, format-specific technical metadata, descriptive metadata for access, non-core preservation metadata
  - Lifecycle management of objects outside repository
  - Rights management: limited to permissions regarding actions taken within repository

PREMIS Maintenance Activity



- Web site:
  - Permanent Web presence, hosted by Library of Congress
  - Central destination for PREMIS-related info, announcements, resources
  - Home of the PREMIS Implementers' Group (PIG) discussion list

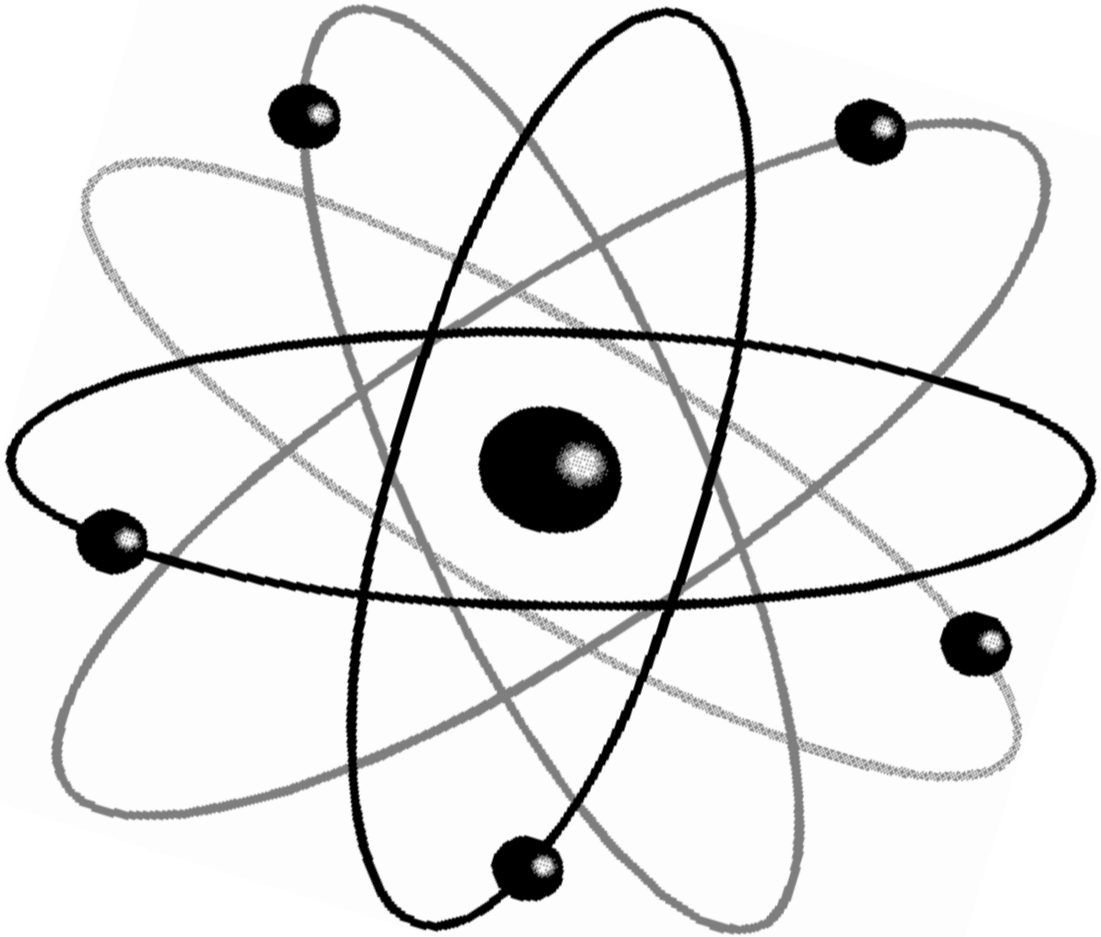
- PREMIS Editorial Committee:
  - Set directions/priorities for PREMIS development
  - Considers proposals for changes
  - Coordinates revisions of Data Dictionary and XML schema

http://www.loc.gov/standards/premis/

Recent Maintenance Activities

- This tutorial and post-iPRES PREMIS Implementation Fair
- Formal revision process (September 2010)
- Revised conformance statement (October 2010)
- *Understanding PREMIS*
  - English, Spanish, Italian, German
- PREMIS in METS
  - PREMIS in METS Toolbox (pim.fcla.edu)
  - Using PREMIS with METS Guidelines
  - Checklist for documenting PREMIS-METS Decisions
- Implementation registry
- PIG list (PIG@LISTSERV.LOC.GOV)

DATA MODEL

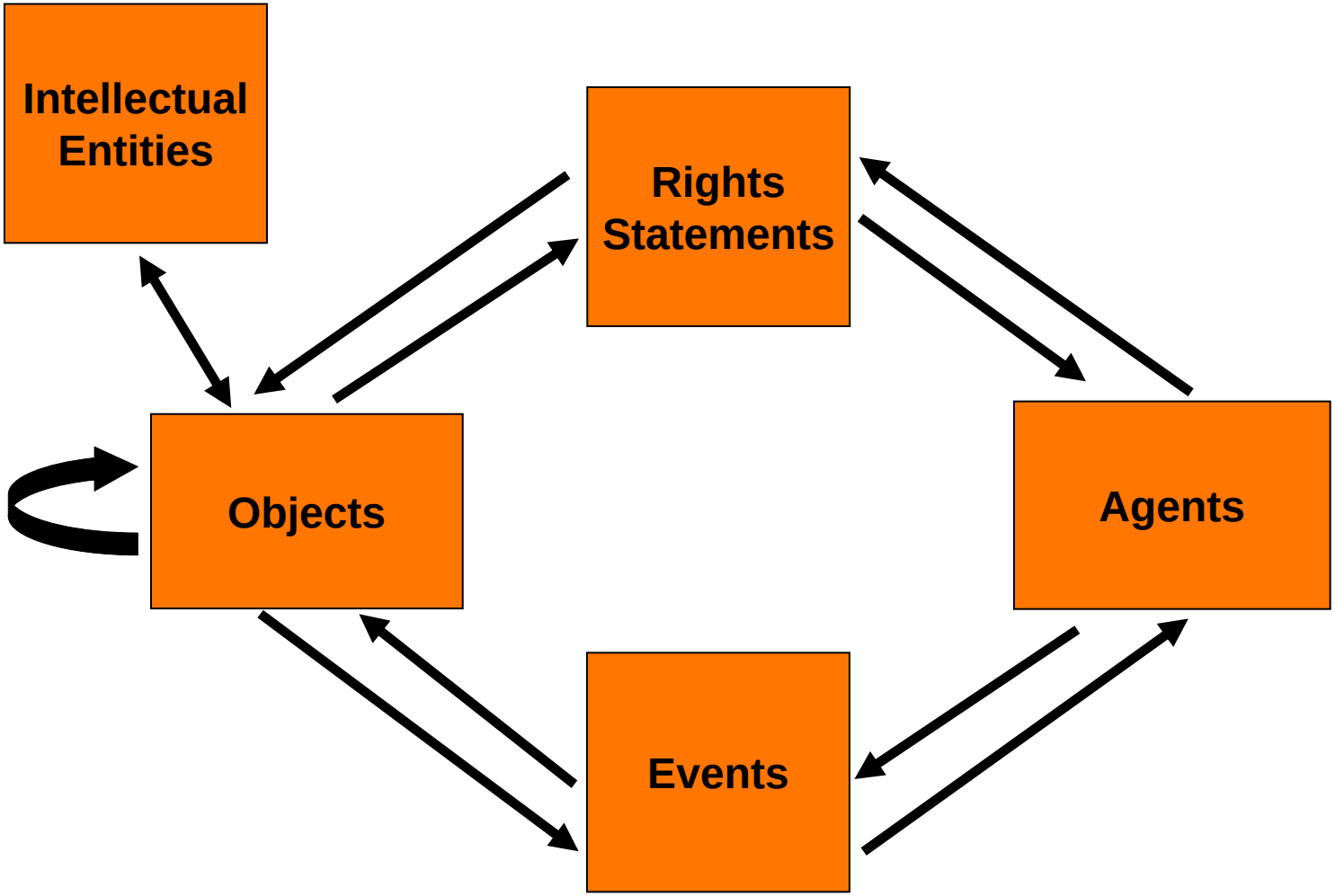


The PREMIS Data Model

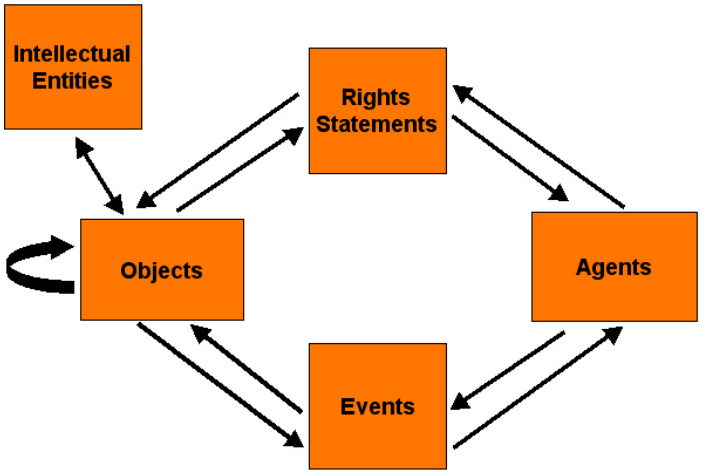
- Data model includes:
  - Entities: “things” relevant to digital preservation that are described by preservation metadata (Intellectual Entities, Objects, Events, Rights, Agents)
  - Properties of Entities (semantic units)
  - Relationships between Entities

- Why have data model?
  - Organizational convenience (for development and use)
  - Useful framework for distinguishing applicability of semantic units across different types of Entities and different types of Objects
  - But: not a formal entity-relationship model; not sufficient to design databases

PREMIS Data Model



Intellectual Entities

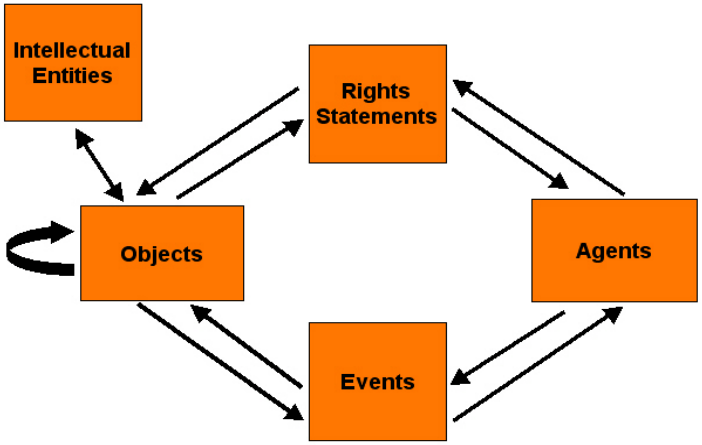


Examples:

- *The Chamber* by John Grisham (an ebook)
- "Maggie at the beach" (a photograph)
- The Library of Congress Website (a website)

- Set of content that is considered a single intellectual unit for purposes of management and description (e.g., a book, a photograph, a map, a database)
- Has one or more digital representations
- May include other Intellectual Entities (e.g. a website that includes a web page)
- Not fully described in PREMIS DD, but can be linked to in metadata describing digital representation **THIS WILL CHANGE**

Objects



Examples:

- a PDF file
- A book composed of several XML files and many images
- TIFF file containing a header and 2 images

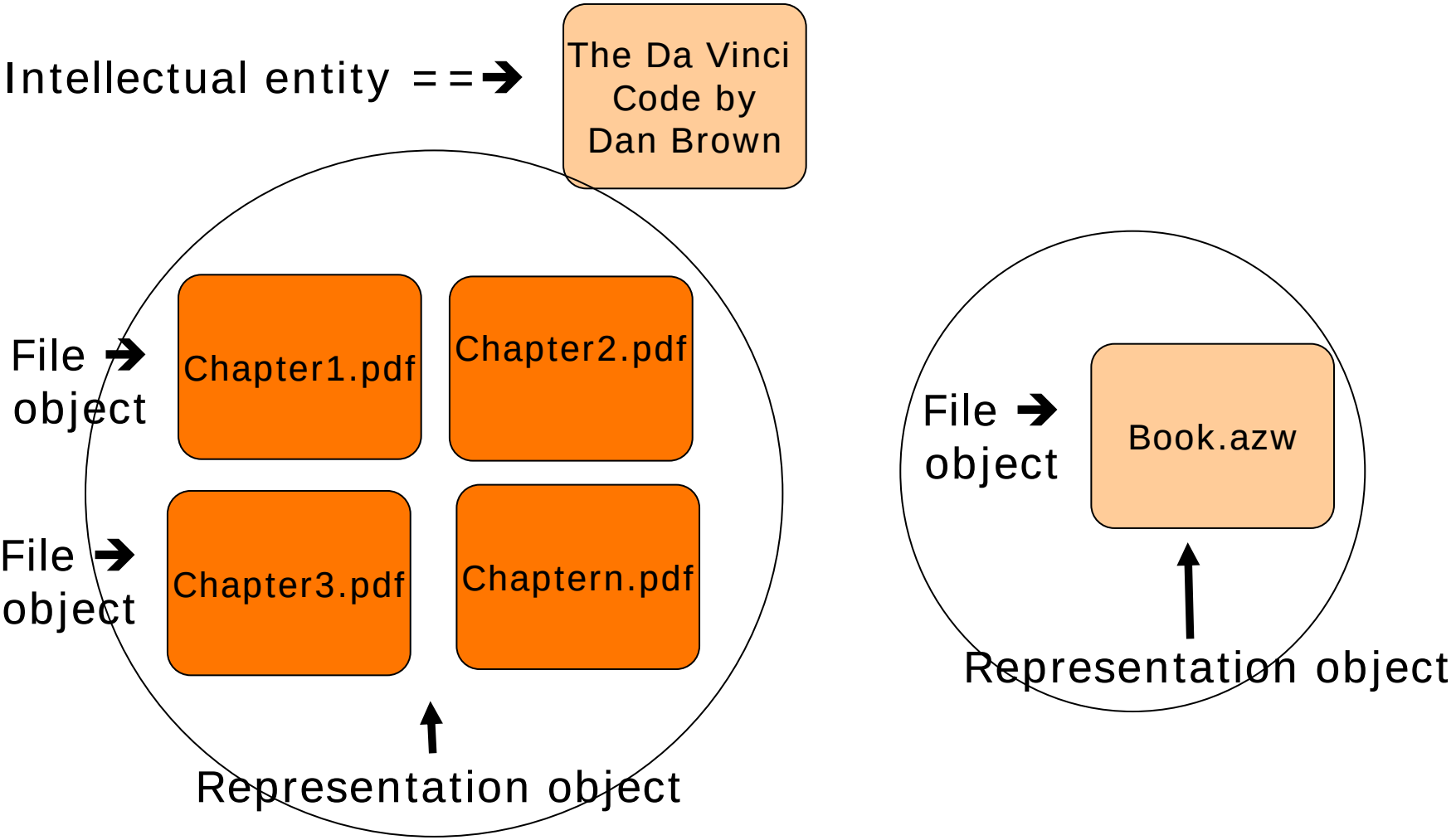
- Objects are what repository actually preserves

FILE: named and ordered sequence of bytes that is known by an operating system

REPRESENTATION: set of files, including structural metadata, that, taken together, constitute a complete rendering of an Intellectual Entity

BITSTREAM: data within a file with properties relevant for preservation purposes (but needs additional structure or reformatting to be stand-alone file)

A book in 2 versions

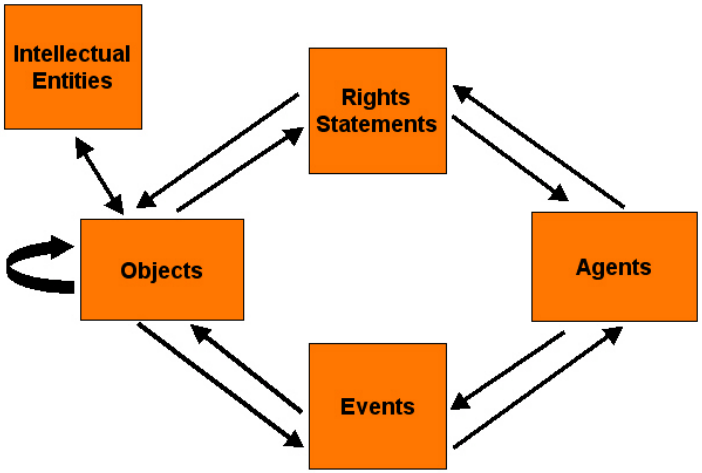


An important aside about Objects

- Repository does NOT have to manage all types of Objects
  - E.g., repository may only manage files, not representations or bit streams.

- The PREMIS DD tells you:
  - **IF** you control at the representation level, these are the semantic units (properties) that pertain to representations;
  - **IF** you control at the file level, these are the semantic units (properties) that pertain to files;
  - **IF** you control at the bitstream level, these are the semantic units (properties) that pertain to bit streams;
  - **AND IF** you control at multiple levels, you need to record relationships between them (more on this soon).

Events

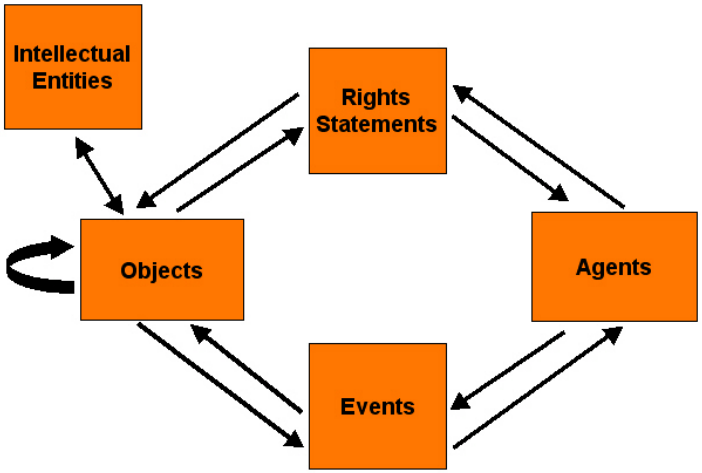


Examples:

- Validation Event: use JHOVE tool to verify that chapter1.pdf is a valid PDF file
- Ingest Event: transform an OAIS SIP into an AIP (one Event or multiple Events?)

- An action that involves or impacts at least one Object or Agent associated with or known by the preservation repository
- Helps document digital provenance. Can track history of Object through the chain of Events that occur during the Objects lifecycle
- Determining which Events are in scope is up to the repository (e.g., Events which occur before ingest, or after de-accession)
- Determining which Events should be recorded, and at what level of granularity is up to the repository

Agents

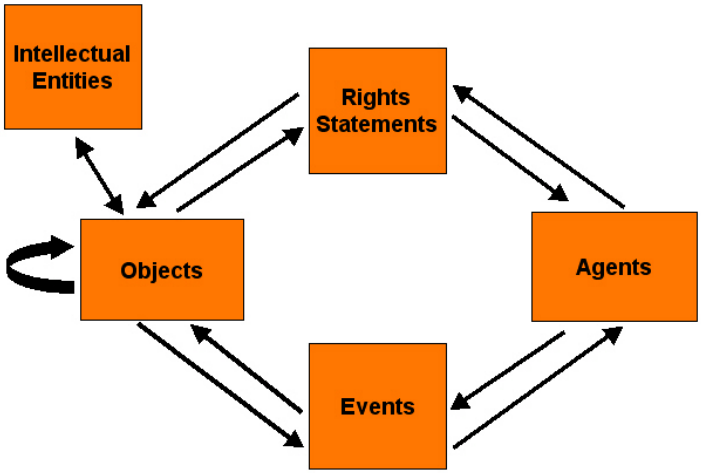


Examples:

- Priscilla Caplan (a person)
- Florida Center for Library Automation (an organization)
- JHOVE version 1.0 (a software program)

- Person, organization, or software program/system associated with an Event or a Right (permission statement)
- Agents are associated only indirectly to Objects through Events or Rights
- Not defined in detail in PREMIS DD; not considered core preservation metadata beyond identification

Rights Statements



- An agreement with a rights holder that grants permission for the repository to undertake an action(s) associated with an Object(s) in the repository.
- Not a full rights expression language; focuses on permissions that take the form:
  - Agent X grants Permission Y to the repository in regard to Object Z.
- Basis for rights may be copyright, license or contract

Semantic units

- A semantic unit is a property of an Entity
  - Something you *need to know* about an Object, Event, Agent, Right
  - Piece of information most repositories need to know in order to carry out their digital preservation functions

- Two kinds of semantic unit:
  - Container: groups together related semantic units
  - Semantic components: semantic units grouped under the same container

- Example:

  ObjectIdentifier [container]

  ObjectIdentifierType [semantic component]

  ObjectIdentifierValue [semantic component]

Semantic units and metadata elements

- A semantic unit is *not* a metadata element
  - Metadata element is an implementation decision (how and whether a semantic unit is recorded in the system)

- Examples:
  - Semantic unit can be recorded in single metadata element, or multiple elements:
    - Example: significantProperties: break up into separate elements for content, “look and feel”, and functionality, or record all in 1 element
  - Semantic unit can be recorded explicitly, or known implicitly
    - Example: IdentifierType: created/assigned internally by repository, assigned to all Objects, so no need to record

- However it is implemented/recorded, a semantic unit should be *recoverable* from archiving system (broadly defined

IDENTIFIERS AND RELATIONSHIPS

Identifiers

- Instances of Objects, Events, Agents and Rights statements are uniquely identified by **Identifiers**

Syntax

- [entity]Identifier
  - [entity]IdentifierType = a specification of the domain in which identifier is unique (e.g. URI, DOI, PURL)
  - [entity]IdentifierValue = the identifier string itself

Example

- ObjectIdentifier
  - ObjectIdentifierType = DRS
  - ObjectIdentifierValue = http://nrs.harvard.edu/urn-3:FHCL.Loeb:sa1

Example

- EventIdentifier
  - EventIdentifierType = DRS
  - EventIdentifierValue = 716593

Some notes on Identifiers

- "IdentifierType" optimally should contain sufficient information to indicate:
  - How to build the value
  - Who is the naming authority
  - Example from previous slide: ObjectIdentifierType = "DRS" (Harvard's Digital Repository Service). Could have also put "URL" (since identifier is unique in both domains) but "DRS" conveys more information.
- If all identifiers are local to repository system, it is unlikely that IdentifierType would be recorded for each identifier in the system
  - BUT should be supplied when exchanging data with others

Relationships

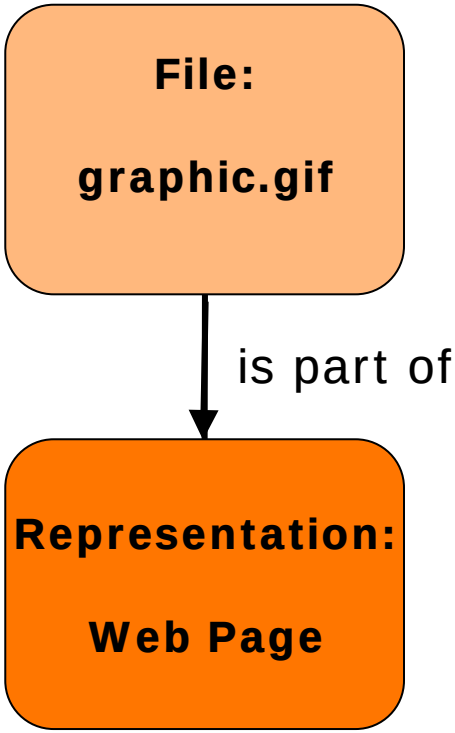
- **Many different types of information relevant to preservation can be expressed as relationships:**
  - e.g., “A is part of B”, “A is scanned from B”, “A is a version of B”
- **PREMIS Data Dictionary supports expression of relationships between:**
  - Different Objects
    - Across same level or different levels
    - Structural: relationships between parts of a whole
    - Derivation: relationships resulting from replication or transformation of an Object
  - Different Entities
- **Relationships are established through reference to Identifiers of other Objects or Entities**

Relationships between Objects: Which, How, Why

- **WHICH Objects are related?**
  - relatedObjectIdentification: type, value
  - relatedObjectSequence: documents "ordered" relationships: e.g., pages, chapters, slide #
- **HOW are the Objects related?**
  - relationshipType: structural, derivation
  - relationshipSubType: "is part of", "is source of", "is derived from"
- **WHY are the Objects related?**
  - Was relationship result of an Event? (e.g., "migration", "replication")
  - relatedEventIdentification: type, value
  - relatedEventSequence: ordered sequence of Events
    - Event 1: Convert Excel spreadsheet to ASCII tab-delimited file
    - Event 2: Convert ASCII file to new spreadsheet format
    - Avoids numerous bilateral format-to-format conversions

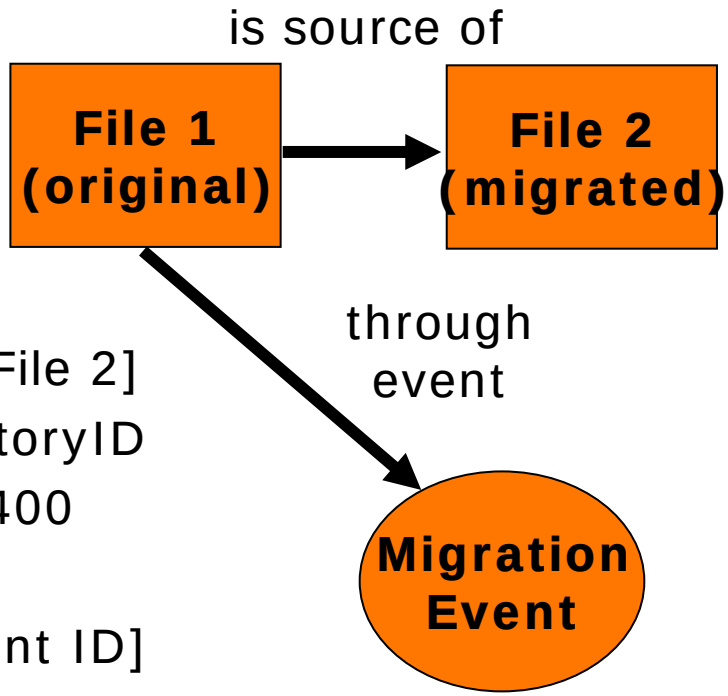
Example: Structural relationship File “is part of” Representation

relationship [part of the description of File]
- relationshipType = structural
- relationshipSubType = is part of
- relatedObjectIdentification [the Web page]
  - relatedObjectIdentifierType = repositoryID
  - relatedObjectIdentifierValue = 0385503954
  - relatedObjectSequence = 0
- relatedEventIdentification [none]

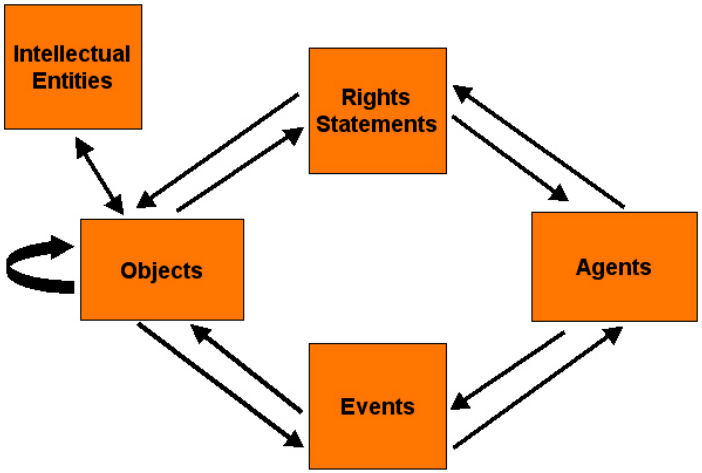


Example: Derivation relationship
File 1 “is source of” File 2 through Migration Event

relationship [part of description of File 1]
- relationshipType = derivation
- relationshipSubType = is source of
- relatedObjectIdentification [identifier of File 2]
  - relatedObjectIdentifierType = repositoryID
  - relatedObjectIdentifierValue = F004400
  - relatedObjectSequence [none]
- relatedEventIdentification [Migration Event ID]
  - relatedEventIdentifierType = repEventID
  - relatedEventIdentifierValue = E0192
  - relatedEventSequence [none]



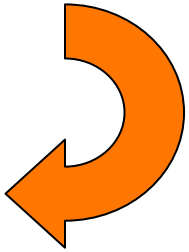
Relationships between different Entities



- **Identifiers are used to link related Entities together**
- **For example, an Object can link to one or more Intellectual Entities, Rights statements, and Events via "linking" semantic units**

Object A is associated with Event 6
The description of Object A contains:

linkingEventIdentifier
linkingEventIdentifierType
linkingEventIdentifierValue



This is some relationship that does not involve another object – for example, a virus check event.

Data dictionary descriptions

Semantic unit	Name that is descriptive and unique. Use externally aids interoperability. Need not be used internally in repository.
Semantic components	If a container, lists its sub-elements. Each component has own entry.
Definition	Meaning of semantic unit
Rationale	Why the unit is needed (if not obvious)
Data constraint	How it should be encoded; ”Container”: an umbrella for two or more; no values given ”None”: can take any form ”Value should be taken from a controlled vocabulary”
Object category	Representation File Bit stream
Applicability	Whether it applies to the category of object
Examples	Illustrative examples of values
Repeatability	Whether it can take multiple values
Obligation	Whether values must be given. ”Mandatory”: something the repository must know independent of how or whether the repository records it. Means mandatory if applicable. If not explicitly recorded, it must be provided in exchange.
Creation /maintenance notes	Information about how values may be obtained or updated.
Usage notes	Information about intended use.

For each level of Object

Sample Data Dictionary entry

Semantic unit	size		
Semantic components	None		
Definition	The size in bytes of the file or bitstream stored in the repository.		
Rationale	Size is useful for ensuring the correct number of bytes from storage have been retrieved and that an application has enough room to move or process files. It might also be used when billing for storage.		
Data constraint	Integer		
Object category	**Representation**	**File**	**Bitstream**
Applicability	Not applicable	Applicable	Applicable
Examples		2038927	
Repeatability		Not repeatable	Not repeatable
Obligation		Optional	Optional
Creation/ Maintenance notes	Automatically obtained by the repository.		
Usage notes	Defining this semantic unit as size in bytes makes it unnecessary to record a unit of measurement. However, for the purpose of data exchange the unit of measurement should be stated or understood by both partners.		