

# XIP and PREMIS

Mark Evans

# What is XIP

- Packaging and Metadata schema used by within SDB / Preservica
- Used to represent all three information packages
  - Includes capture of structural, descriptive and technical metadata
- Relatively long history
  - First developed in 2002 prior to PREMIS
  - Major revamp in 2005 just as PREMIS was released
  - Incorporated elements from PLANETS
- Can export XIP as METS / PREMIS
  - Some mappings are a little awkward / lossy

# XIP Structure

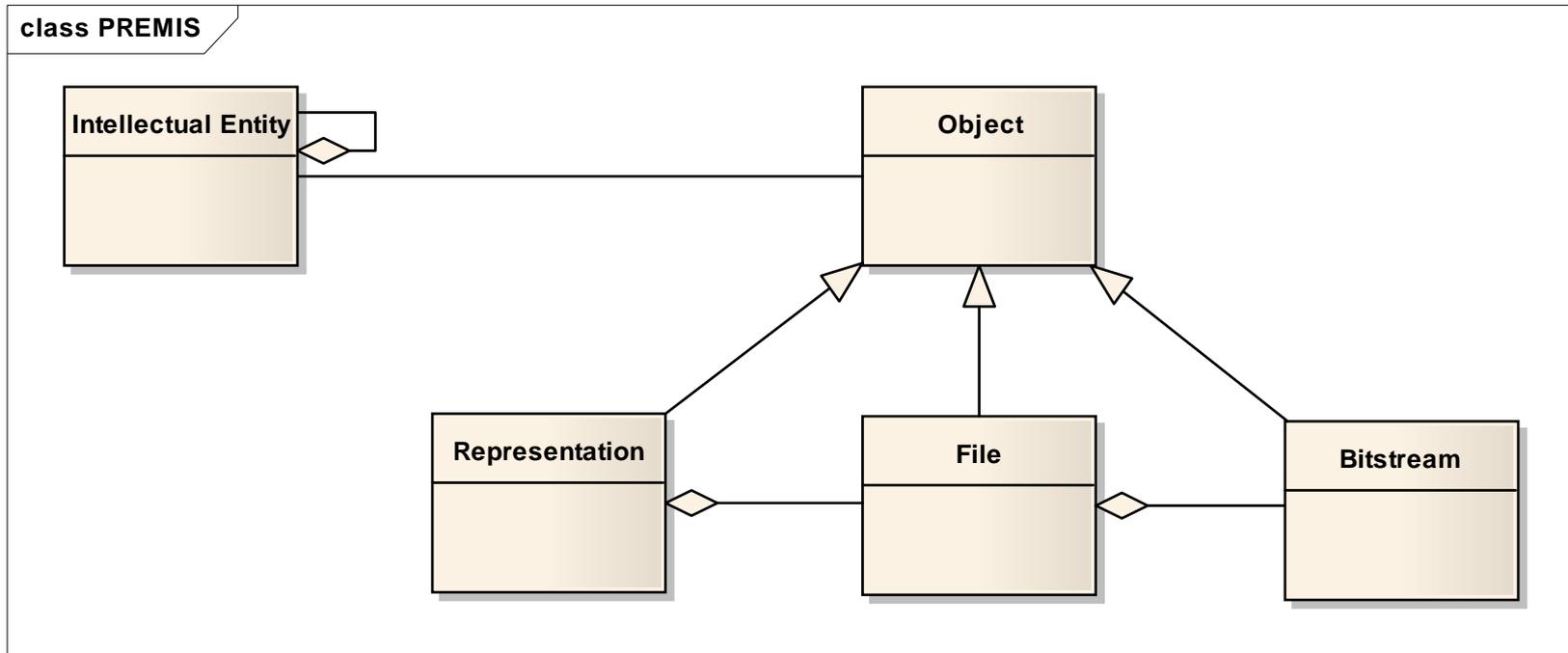
- Informational objects
  - Collection
  - Deliverable Unit – An IE that can be deposited or delivered
- Digital objects
  - File
  - Embedded Bytestream
- Manifestation
  - Similar to PREMIS “Representation”
  - Allows multiple representations of the same intellectual object
  - Required for preservation – File migration

# XIP Metadata

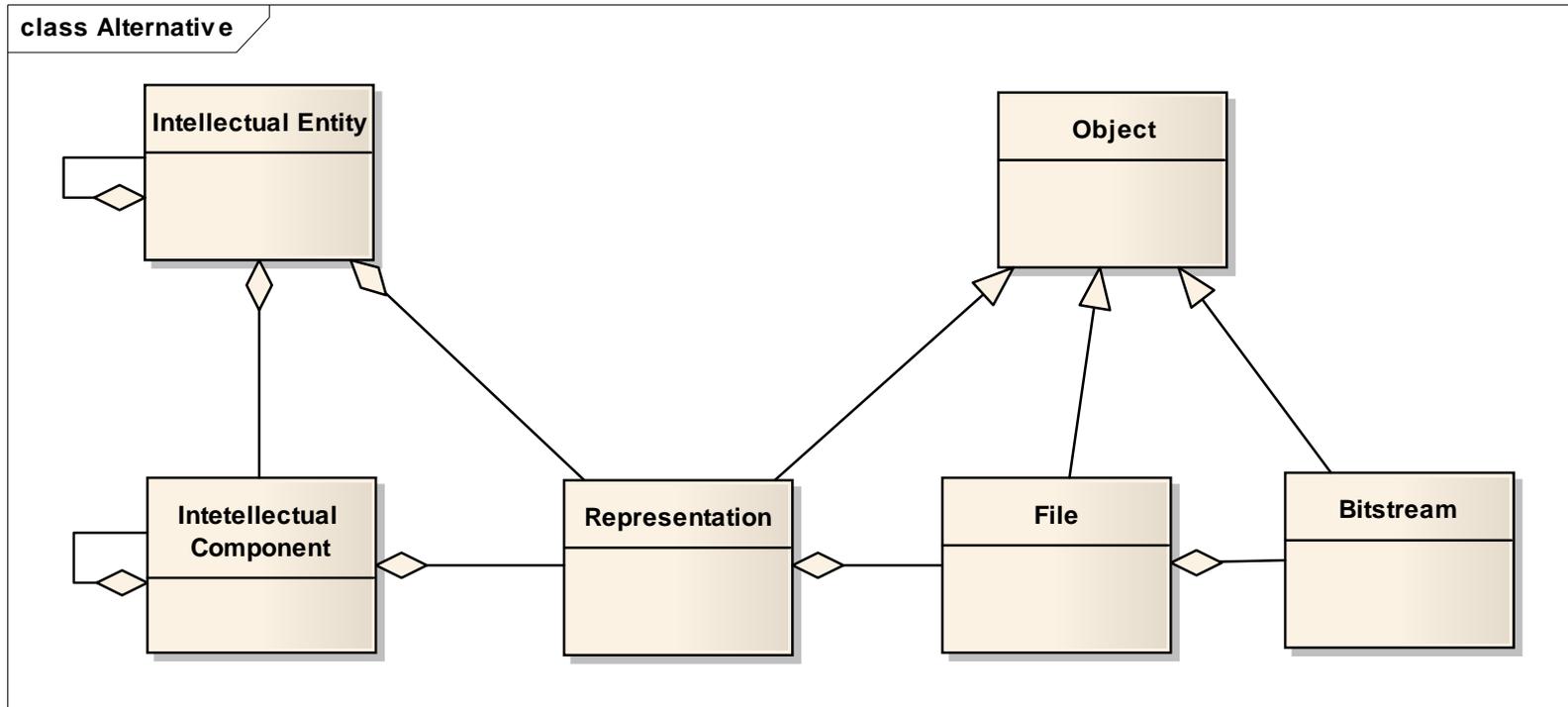
- Descriptive Metadata
  - Applies to all entities
  - Can embed any schema – e.g Dublin core
- Technical Metadata
  - Applies to files and bytestreams
  - Link to format registry content via PRONOM ID
  - Includes extracted properties
- Event metadata
  - Held explicitly within the entity metadata

# Comparison of PREMIS 2, SDB and PREMIS 3

# PREMIS 2 – Partial Model



# SDB Partial Model



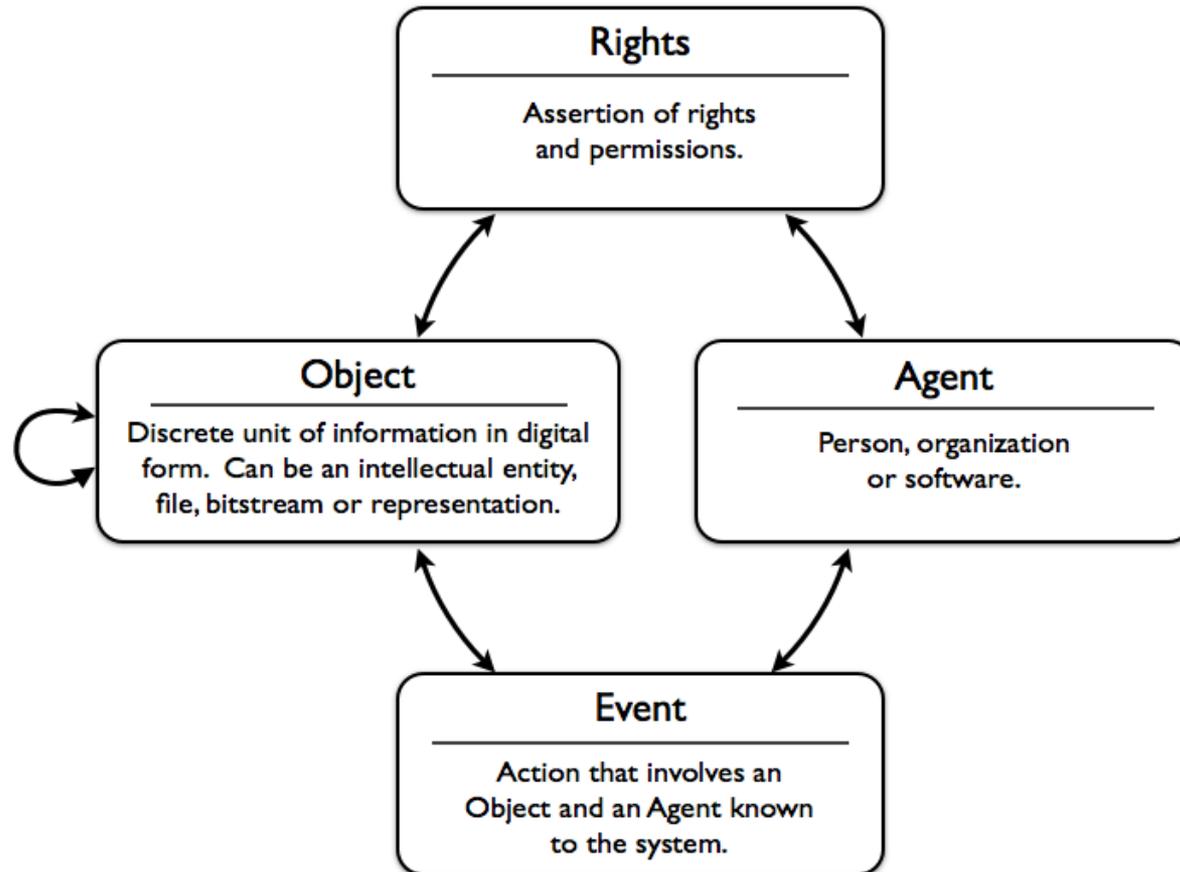
More complex than original PREMIS would allow

Break IE into fundamental components

# Other Differences

- Event model in XIP is not as generalized
  - Some events are implicitly tied to another entity
- Rights and Agent model is simplistic in XIP
- Some information is expected to be else where
  - E.g. File format information in a registry
  - E.g User information in a directory
- XIP provides additions to support preservation planning and action within SDB

# PREMIS 3 Model



# Comparison Conclusions

- PREMIS 3 generalizes model
- Object is now one of:
  - Intellectual Entity
  - Representation
  - File
  - Bitstream
- Allows flexibility we need so we can now map to PREMIS better
  - E.g. We have a potential way of representing components in PREMIS.

# Active Preservation

# Active Preservation Overview

- Distinguish physical & conceptual
- Conceptual (Intellectual Entity)
  - We distinguish:
    - “Deliverable Unit” (can be delivered and/or described by cataloguer)
    - “Component” (might not be deliverable or have description but we need to detect its presence & ensure it is preserved)
- Physical:
  - Representation (of both Deliverable Units & Components)
  - File
  - Bitstream

# Active Preservation - Migration

- Detect issue in File / Bitstream property (e.g., format)
- Hence, find Deliverable Unit Representations “at risk”
- Whole Deliverable Unit Representation is “migrated”
- Do this by migrating all of its Component Representations
- For each Component Representation:
  - Must still exist
  - Must retain “significant properties” (within tolerance)
  - Must retain links

# Active Preservation – Migration II

- Worth Noting

- No requirement to preserve file structure through a migration
- Some Component Representations unchanged
- Some only changed via links (e.g., update HTML file to link to new image)

- Generate a “Transformational Entity”

- Represents the act / event of planning and migration
- Objects considered at risk
- The Action to take / taken
- Related to pre and post DU manifestations

# Active Preservation - Challenge

- The Challenge:
  - Persist the Transformational Entities (between workflow steps), as the plan is being created and actioned
  - Not sure how to do these in PREMIS?
  - Closes PREMIS entity is “Event”
  - SDB Transformational Entities are lost when we export to PREMIS

## Topic 3 - Verbosity

- Really an artefact of export?
- When export need to be explicit about:
  - Identifier types
  - Format registry
  - Creating application, environment, software and hardware entities?
  - etc.
- This makes PREMIS export much bigger than internal metadata (where all implicit)
- Moves afoot in version 3 to reduce this

# Summary

- PREMIS has been evolving
- Tessella are engaged in this process
- We are committed to revisit our export functionality to be compatible with forthcoming versions