

# HEG-796-22-030

PREMIS et SHACL

Jan Krause-Bilvin

2022-04-11

## Thème de cette session

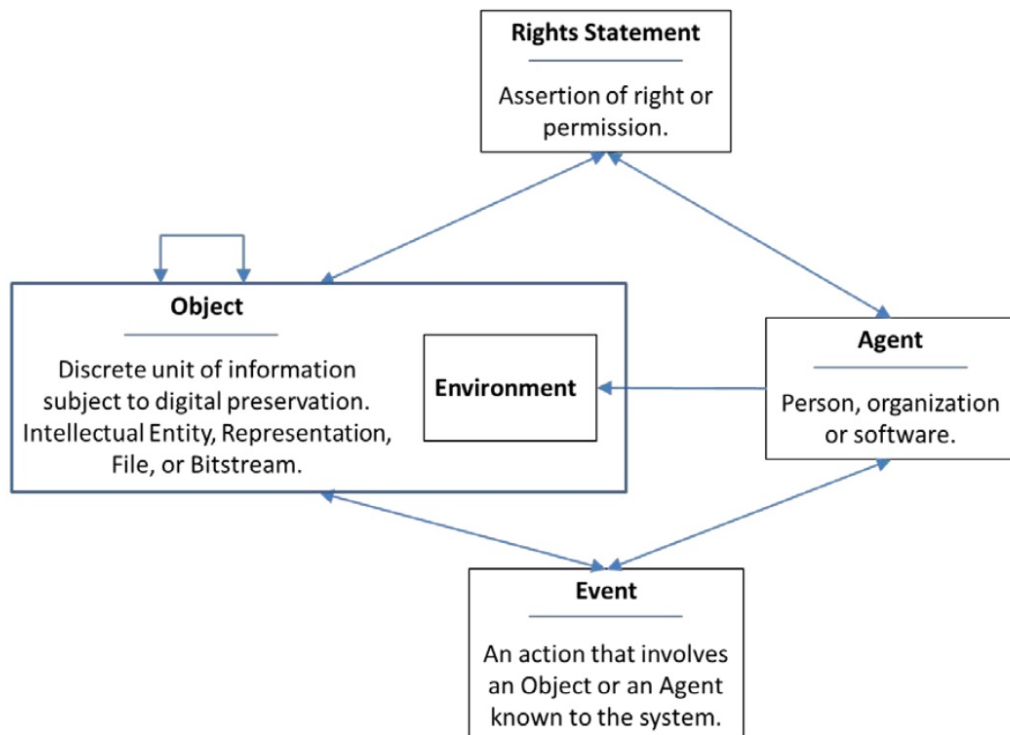
- Ontologie de préservation: ([PREMIS](#))
- Schémas de graphes RDF ([SHACL](#))

## PREMIS

PREservation Metadata : Implementation Strategies (PREMIS) permet de représenter:

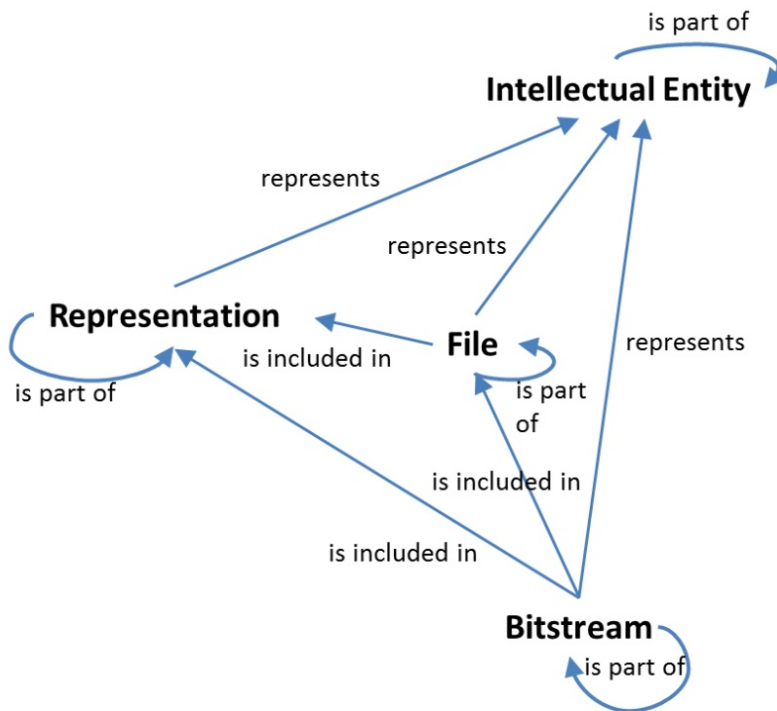
- les objets (p.ex. records),
- les événements de préservation,
- les agents (personnes, logiciels) impliqués dans ces événements,
- les droits.

en RDF.



PREMIS overview

Une représentation fine des objets numérique est disponible.



PREMIS overview

Exemples de concepts PREMIS:

- premis:hasMessageDigest (checksum)
- premis:signature (signature numérique)
- premis:hasSize (taille en octets)
- premis:originalName (nom original)
- premis:rights (droits)
- premis:formatRegistry (format de fichier)
- premis:creatingApplication (application)
- premis:environmentDesignation (système)
- premis:inhibitors (inhibiteurs t.q. DRM)
- premis:hasCompositionLevel (composition)

PREMIS peut être combiné aux ontologies descriptives pour assurer la préservation numérique. Par exemple:

PREMIS	RiC-O
-	RecordSet RiC-E03
Intellectual entity	Record RiC-E04
Representation	Instantiation RiC-E06
File	-
Datastream	-

Mais comment faire en pratique?

## SHACL (Core)

[SHape and Constraint Language](#) (W3C)

- Il s'agit d'un langage de validation de graphe RDF.
- Les graphes sont composés de noeuds (ensembles de triplets).
- Validation porte sur la structure et le contenu des noeuds.

### Exemple

```

ex:Alice
  a ex:Person ;
  ex:ssn "987-65-432A" .

ex:Bob
  a ex:Person ;
  ex:ssn "987-65-432B" ;
  ex:birthDate "1971-07-07"^^xsd:date ;
  
```

SHACL

```
ex:PersonShape
  a sh:NodeShape ;
  sh:targetClass ex:Person ;      # toutes les personnes
  sh:property [                  # _:b1
    sh:path ex:ssn ;             # contraintes ex:ssn
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:datatype xsd:string ;
  ] ;
  sh:property [                  # _:b2
    sh:path ex:birthDate ;
    sh:maxCount 1 ;
    sh:datatype xsd:date ;
  ] ;
```

---

## Nous allons nous focaliser sur

- Nombre d'occurences:
  - ***sh:minCount*** , ***sh:maxCount***
- Type de noeud:
  - ***sh:NodeKind sh:IRI*** , ***sh:NodeKind sh:BlankNode***
- Type de données:
  - ***sh:datatype xsd:date*** , ***sh:datatype xsd:string***
- Format
  - ***sh:maxLength 50***
  - ***sh:pattern*** `“^d{3}\.d{4}\.d{4}\.d{2}”`

---

## Mode de validation fermé

Pour un noeud donné, le mode fermé (***sh:closed true***), requiert que chaque triplet satisfasse au moins une condition énoncée.

Par défaut, le mode est ouvert. En d'autre termes, les triplets non concernés pas les conditions sont ignorés.

---

## Démonstration

---