

HEG-796-22-050

Intégration et discussion

Jan Krause-Bilvin

2022-05-17

Récapitulation et intégration

Thématiques couvertes dans ce cours

Modélisation de:

1. la description d'objets (Ontologie RiC)
 2. la préservation d'objets (Ontologie PREMIS)
 3. l'association d'ontologies et contraintes aditionnelles, avec validation formelle (SHACL)
 4. l'empaquetage de ressources RDF et binaires et de leur versionnage (containers LDP, Fedora Commons)
 5. la préservation des containers LDP versionnés dans des AIP au sens la norme OAIS (OCFL, Fedora Commons)
 6. valorisation d'objets préservés (cours suivant)
-

1. Description Ontologie RiC

- Record ressources : collections, fonds, séries, dossiers, documents
- Instanciations : item ou article à propoient dit (fichiers, objets physiques)
- Événements : création, clôturée, etc.
- Lieux : géographie
- Dates : ponctuelles, liste ou plages
- Agents : groupes, personnes morales ou physiques, agents logiciels

=> *ontologie du domaine des archives*

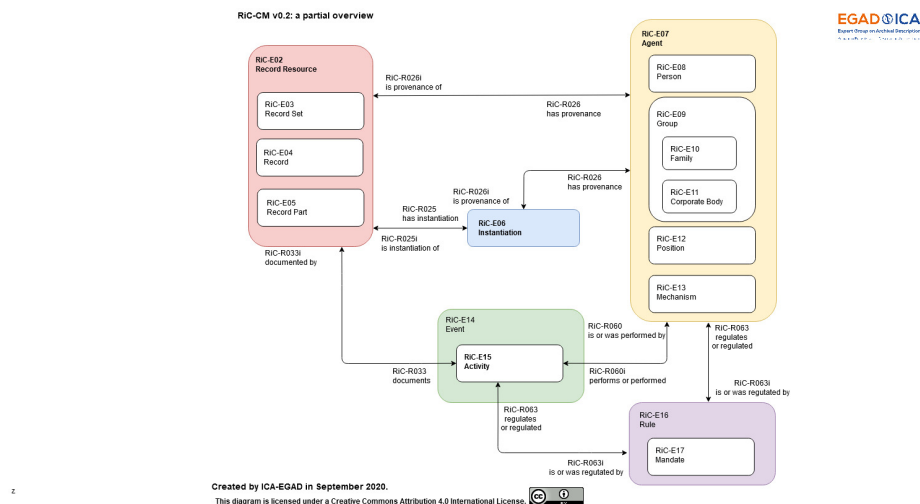


FIGURE 1 – RiC overview

Exemple:

```

ex:record371          a          rico:Record ;
                      rico:title   'Photo du Cervin' ;
                      rico:hasInstantiation  ex:instantiation472 .
ex:instantiation472   a          rico:Instantiation ;
                      rico:name     '20220525_123458.jpg' .
  
```

2. Métadonnées de préservation (PREMIS)

PREMIS peut être combiné à RiC-O:

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

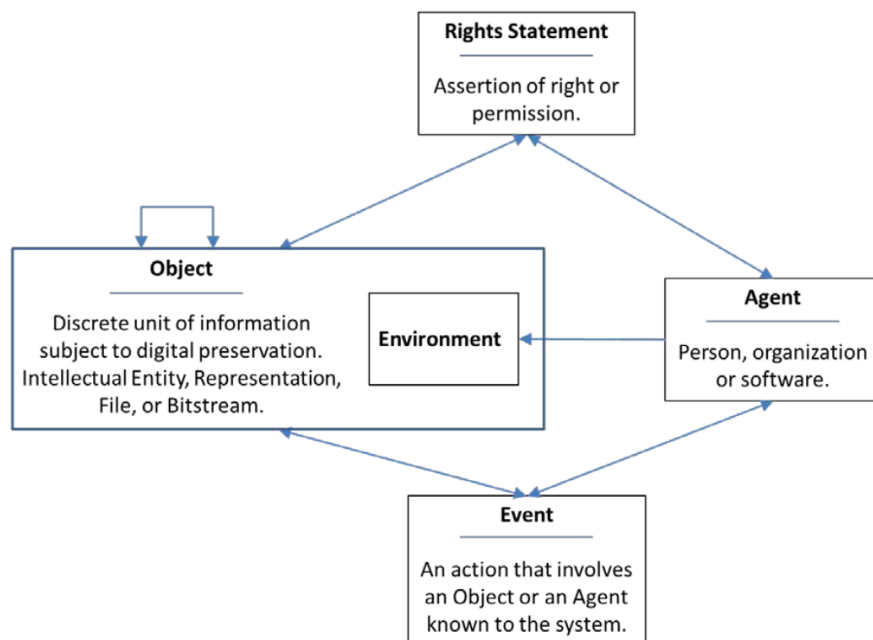


FIGURE 2 – PREMIS overview

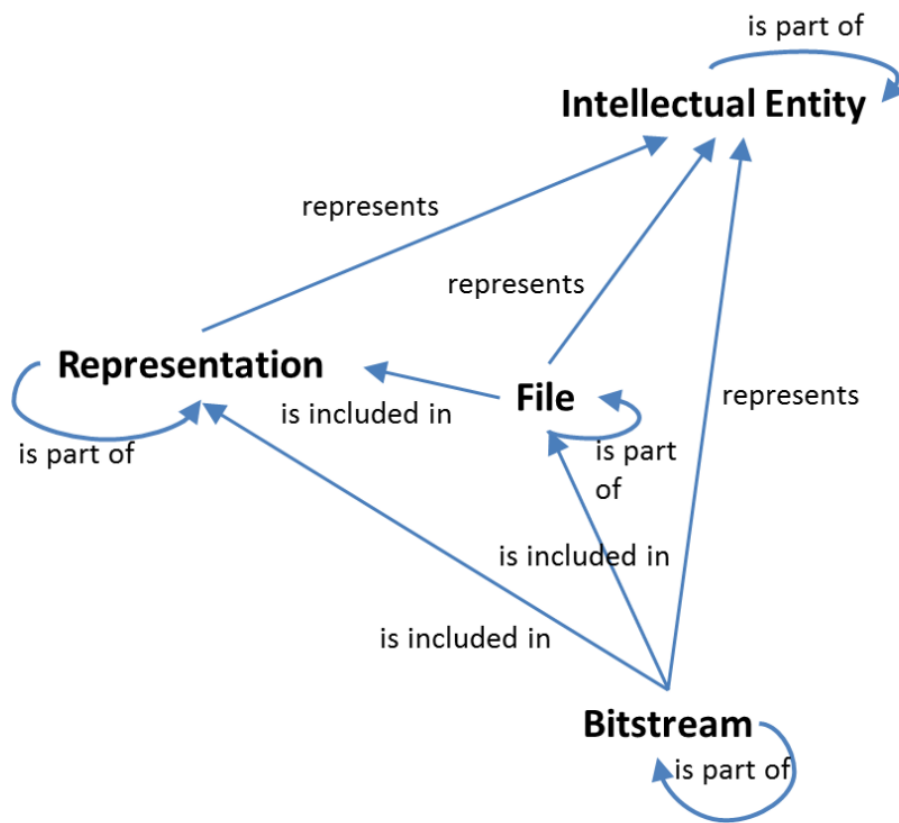


FIGURE 3 – PREMIS overview

```

ex:instantiation472    a          rico:Instantiation ;
                       a          premis:Representation;
                       rico:title  'Captured with Fariphone 4' ;
                       premis:includes ex:file9642 .
ex:file9642            a          premis:File ;
                       premis:originalName  '20220525_123458.jpg' ;
                       premis:messageDigest  '16a993c472d589f7bc36922...';
                       premis:messageDigestAlgorithm  'SHA512';
                       premis:size          1023 ;
                       premis:formatRegistryKey 'fmt/432' ;
                       premis:formatRegistryName 'PRONOM' .

```

3. Association d'ontologies et contraintes (SHACL)

- définition de l'articulation des ontologies (ex. combinaison RiC et PREMIS)
- contraintes personnalisées à souhait (ex. minimum un agent-auteur avec numéro AVS valide)
- validation fermée ou ouverte à choix (permet de la souplesse... ou pas)
- permet d'étendre à volonté l'ontologie descriptive (ex. RiC+SKOS+EbuCore ou DublinCore+SKOS+RDAU)

=> *respect d'un schéma rigoureux, fondamental pour la préservation*

Extension de la description

Autres ontologies descriptives à ne pas rater:

- **DublinCore** : **Element Set** et **DCMI Terms**
 - **RDA-U** : Research Data Alliance - nombreuses propriétés concernant les records et agents
 - **EBU-core** : Eurovision data model - multimedia
 - **IFLA-LRM** : Library Reference Model - nouvelle évolution après FRBR
 - **CIDOC-CRM** : Musées
-

```

ex:InstantiationShape
  a sh:NodeShape ;
  sh:targetClass rico:Instantiation ;
  sh:property [
    sh:path rdfs:type ;
    sh:class premis:Representation ;
  ] ;
  sh:property [

```

```

    sh:path premis:includes ;
    sh:minCount 1 ;
    sh:NodeKind sh:IRI ;
  ] .

```

4. Empaquetage des ressources RDF et binaires (LDP)

- Containers LDP permettent de conditionner les ressources de façon à ce qu’elles coresspondent aux objets à archiver (dossiers, documents, etc.).
- Gestion des versions des objets (RFC 7089, Memento)
- Normalisation de la manipulation des containers (création, modification, etc.)

=> *Adéquation avec les objets métier à archiver et interopérabilité*

LDP BasicContainer:

```

ex:record371      a          rico:Record ;
                  rico:title   'Photo du Cervin' ;
                  rico:hasInstantiation ex:instantiation472 ;
                  ldp:contains ex:instantiation472 .

```

LDP BasicContainer:

```

ex:instantiation472 a          rico:Instantiation ;
                   a          premis:Representation;
                   premis:includes ex:file9642 .
...

```

5. Préservation (OAIS, OCFL)

Chaque objet OCFL peut préserver un ou plusieurs containers LDP. OCFL a cinq objectifs principaux:

- Complétude (disaster recovery)
 - Parsabilité (humains et machines)
 - Robustesse (erreur, corruption, migrations)
 - Versionnage (historique des objets)
 - Diversité de stockage (multi-infrastructure et migrations)
-

Créer un “archival unit” via l’API

```
import requests
```

```

url = 'http://localhost:8080/rest/records/acv/D9999'
headers = {"Content-Type": "text/turtle",
           "Link": '<http://fedora.info/definitions/v4/repository#ArchivalGroup>;rel="type"'
auth = ('fedoraAdmin', 'fedoraAdmin')
data = """ <rico:title>          'Ceci est le titre'.
           <rico:scopeAndContent> 'Voilà la description'.
           """

r = requests.put(url, auth=auth, data=data.encode('utf-8'), headers=headers)
print( 'Status:', r.status_code )
print( r.text )

```

Noter la partie “link” dans les headers.

Intégration

Animation d'intégration

Discussion

La combinaison de ces standards et outils permet:

1. Généraliser la description / catalogage
 2. Préservation à long terme des objets
 3. Une excellente interopérabilité (cf. cours suivant)
-

1. Généralisation de la description

- Richesse des ontologies “métier”: **LOD cloud**
 - Interconnexions (owl:sameAs) entre les ontologies
 - Les contraintes d'accès pour les usagers tombent (ISAD(G): accès selon la structure arborescente des fonds imposée)
-

Focus sur la structure

P.ex. RiC, et les technologies du Web sémantique en général, offre un accès multi-dimensionnel (pas seulement selon un arbre):

- Les “records sets” ne sont plus limités à une structure arborescente bidimensionnelle.
 - D’abord, l’arbre peut changer dans le temps => structure tridimensionnelle (demo dans Fedora).
 - Plusieurs regroupements intellectuels peuvent être faits et combinés (rico:proxy), exemple de [BodmerLab](#) et d’[Europeana/Histoires](#) et .
 - L’accès par d’autres types d’objets (agents, sujets, lieux, fonctions, etc.) est facilité.
-

2.1. Préservation : Périmètre des objets

- Les containers LDP permettent de regrouper le RDF relatif aux objets à préserver (dossiers, documents ou autre).
 - OCFL permet de gérer les diverses versions des objets à préserver dans les AIP de façon fiable et portable (voir de plus les 5 objectifs).
 - Le concept d’*unité archivistique* (*archival unit*) permet de regrouper les objets qui vont ensemble dans un seul AIP (p. ex. un dossier et ses documents).
-

2.1. Préservation : format pérenne

Le RDF est basé sur le concept sujet-objet-prédicat:

- standard du W3C
- structure universelle et de ce fait épargnée par l’obsolescence technologique
- très utilisée, dans de nombreux domaines (informatique, biologie, etc.)

=> *format standard, simple et très utilisé => pérenne*

3. Interopérabilité technique

Est très élevée, c’est en effet le but premier du Web sémantique.

Ceci sera développé dans la prochaine et dernière session de ce cours.

Questions / réponses
