Value Chain Ontologies for Intellectual Property Objects

Marc Gauvin
NetPortedItems / sDae
Jaime Delgado
NetPortedItems / DMAG-UPC
http://www.digitalmediavalues.com/

Introduction

• Need for standard semantics and syntax for the management of Intellectual Property (IP).
• Different options:
  – MPEG-21, ODRL: Data dictionaries (semantics), Rights Expression Languages (syntax).
  – DMP (Digital Media Project): RRD new approach.
• Establish a digital formal representation of generic IP Entities (DMP Creation Model).
• Answer to DMP Call: sDae, NetPortedItems.

Contents

• Introduction
• DMP Creation Model
• IP objects and value chain
• Ontology formalisation
• An application
• Conclusions and future work

The Digital Media Value Chain

Digital Media Project (DMP), etc.:
**Creation Model: basic schema**

**IP Objects relations (I)**

- **Work**
  - Neither Depends or Uses any other Object.
  - Has one-to-many relationship to Adaptation, Manifestation, Manifestation Copy, Instance and Instance Copy.

- **Adaptation**
  - Depends on a Work.
  - Uses Manifestation of Work.
  - Has a one-to-one relationship to Manifestation, Manifestation Copy, Instance and Instance Copy.

**IP Objects relations (II)**

- **Manifestation**
  - Depends on Work or Adaptation.
  - Has one-to-one relationship to Adaptation and/or Work, Instance and Copy.

- **Instance**
  - Uses Manifestation of Work.
  - Has one-to-many relationship to Instance Copy.
Expressing the IP Value Chain

<table>
<thead>
<tr>
<th>Action Sequence</th>
<th>Rights Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produce-Use-Depend-Depend =&gt; Adaptation Instance Copy</td>
<td>Require Rights for:</td>
</tr>
<tr>
<td></td>
<td>• Adaptation of Work</td>
</tr>
<tr>
<td></td>
<td>• First fixation of Adaptation</td>
</tr>
<tr>
<td></td>
<td>• Copy of Adaptation Instance</td>
</tr>
<tr>
<td>Produce-Use-Depend =&gt; Work Instance Copy</td>
<td>Require Rights for:</td>
</tr>
<tr>
<td></td>
<td>• First fixation of Work</td>
</tr>
<tr>
<td></td>
<td>• Copy of Work Instance</td>
</tr>
<tr>
<td>Produce-Depend-Depend =&gt; Adaptation Manifestation Copy</td>
<td>Require Rights for:</td>
</tr>
<tr>
<td></td>
<td>• Adaptation of Work</td>
</tr>
<tr>
<td>Produce-Depend =&gt; Work Manifestation Copy</td>
<td>No Required Rights</td>
</tr>
</tbody>
</table>

Ontologies

An **Ontology**: Data model, represents a domain, allows to reason about the objects in that domain and the relations between them.

**CONCEPTS**:
- **Classes**: sets, collections, or types of objects. *E.g. “Work”*
- **Attributes**: properties, features, characteristics, or parameters that objects can have and share. *E.g. “Work” has a title*
- **Relations**: ways that objects can be related to one another. *E.g. “Adaptation derivesFrom Work”*
- **Individuals**: the ground level objects. *E.g. “Les miserables”*

Ontology formalisation

- The **RRD** (Represent Rights Data) Ontology has been expressed in **OWL** (Ontology Web Language).
- Main classes implemented in OWL are:
  - Objects: IP Entities are subdivided into instance, adaptation, manifestation and copy.
  - Agents/Roles: No further sub-classing.
  - Actions: Grouped into those that apply specifically to digital objects and those which may or may not.

Creation Model Ontology (I):

**Basic elements**

Basic classes:
- Actions: *play, edit, …*
- Agents/Roles: *end-user, creator, …*
- IP Entities: *work, manifestation*

Basic relations:
- ownRights
- requiredRights
- operateOn
Creation Model Ontology (II): IP Entities

Creation Model Ontology (III): Agents & Actions

- Agents/Roles:
  - Distributor, Creator, Adaptor, EndUser, Instantiator

- Actions:
  - For analogue items: fixation, distribution…
  - For digital items: copy, delete…

Creation Model Ontology (IV): Example of relations
An Application: License Processors

API for ontology use:

- **RightsVerification**
  - This function determines whether or not an agent can perform an action on a given object

- **GetAllowedRights**
  - This function determines what actions an agent can perform on a given object

- **GetRightsHierarchy**
  - This returns all parent actions of a given action according to the RRD Ontology Hierarchy

Conclusions and future work

- Finalise formalisation of RRD DMP proposal.
- API implementation.
- Extension from *Creation* aspects to *Distribution* aspects.
- Integration in other applications (AXMEDIS European Project, etc.)