Formalising ODRL Semantics using Web Ontologies

Roberto García, Rosa Gil, Isabel Gallego and Jaime Delgado

***Abstract*—In order to move Digital Rights Management to the Internet, a common rights expression language is needed. ODRL (Open Digital Rights Language) is one of the proposed solutions. It is based on a XML language and thus it just formalises the language syntax, while language semantics are specified informally. Actually, ODRL seems quite complete and generic enough to cope with such a complex domain. However, the problem is that it has such a rich structure that it is difficult to implement. In our opinion, it lacks formal semantics that would help ODRL applications development.**

As the application context is the Web, our approach to formalise ODRL semantics is based on semantic web ontologies. Firstly, ORDL has been moved to the Semantic Web space using XML Schema to OWL and XML to RDF tools. This provides some simple semantics. In order to refine them, the resulting ODRL ontologies have been connected to IPROnto, a result of previous research.

IPROnto, Intellectual Property Rights Ontology, models the IPR core concepts for creation, intellectual property rights and the basic kinds of actions that operate on intellectual property. It enables semantics-aware IPR applications that benefit from semantic queries, in contrast to the difficulties that emerge from the use of syntactic queries when the information space is as complicated as in the IPR field. Moreover, specialised reasoners can be used for license checking and retrieval. All these advantages have been propagated to ODRL thanks to this mapping.

***Index Terms*—Copyright protection, Digital Rights Management, Knowledge representation, Ontology**

I. Introduction

THE amount of digital content delivery in the Internet has made Web-scale Digital Rights Management (DRM) a key issue. Traditionally, DRM Systems (DRMS) have deal with this problem for bounded domains. However, when scaled to the Web, DRMSs are very difficult to develop and maintain. The solution is interoperability of DRMS, i.e. a common framework for understanding that defines a shared rights expression languages and its associated vocabulary.

This work was supported in part by Agent Web, Spanish administration TIC 2002-01336, and VISNET, European Commission IST FP6 Network of Excellence (http://www.visnet-noe.org).

Roberto García, Rosa Gil and Jaime Delgado are with the Technology Department, Universitat Pomeu Fabra, Barcelona, E-08003 Spain (e-mail: roberto.garcia, rosa.gil, jaime.delgado@upf.edu).

Isabel Gallego is with the Computer Architecture Department, Universitat Politècnica de Catalunya, Barcelona, E-08034 Spain (e-mail: isabel.gallego@ac.upc.edu).

ODRL (Open Digital Rights Language – http://odrl.net) [1] is one possible approach to that. It is a XML language defined by two XML Schemas. The first XML Schema defines the language syntax and a basic vocabulary. The second XML schema is called the Data Dictionary. It provides the complete vocabulary with textual definitions and a lightweight formalisation of the vocabulary terms semantics as an XML Schema.

ODRL seems quite complete and generic enough to cope with such a complex domain. However, the problem is that it has such a rich structure that it is difficult to implement. It is rich in the context of XML languages and the "traditional" XML tools like DOM or XPATH. There are too many attributes, elements and complexTypes, see Table 1, to deal with.

Table 1. Number of named XML Schema primitives in ODRL

	ODRL	
Schema	EX-11	DD-11
xsd:attribute	10	3
xsd:complexType	15	2
xsd:element	23	74
Total	127	

For instance, consider looking for all constraints in a right expression that apply to how we can access the licensed content. This would require so many XPATH queries as there are different ways to express constraints. ODRL defines 23 constraints: industry, interval, memory, network, printer, purpose, quality… This amounts to lots of source code, difficult to develop and maintain because it is very sensible to minor changes to the ODRL specification. Fortunately, there is a workaround hidden in the language definitions.

As we have said, there is the language syntax but also some semantics. The substitutionGroup relations among elements and the extension/restriction base ones among complexTypes encode generalisation hierarchies that carry some lightweight, taxonomy-like, semantics.

For instance, all constraints in ODRL are defined as XML elements substituting the o-ex:constraintElement. The difficulty is that although XML Schemas provide this information, it remains hidden when working with instance documents of this XML Schemas.

Moreover, there are more complex semantics encoded in the textual definitions of the Rights Data Dictionary. They are needed each time a programmer is developing an ODRL application and thus they must be "manually" interpreted repeatedly.

Our idea is to make the ODRL semantics explicit in order to exploit ODRL hidden semantics and to attach more complex formalisations that facilitate ODRL applications implementation. This objective can be accomplished using ontologies and we have already tested it in the context of rights expression languages, concretely for the formalisation of the MPEG-21 Rights Data Dictionary semantics [2].

Ontologies are formalisations of a shared conceptualisation. They are formal so they provide the required semantics in a machine-readable form. They can be used to provide the required definitions of the rights expression language terms in a formal form. Thus, from the automatic processing point of view, a more complete vision of the application domain is available and more sophisticated processing can be carried out.

In the Web context, ontologies are promoted by the Semantic Web initiative [3] as a tool for Web-wide semantics-enabled processing. We have taken the Semantic Web approach because it is naturally prepared for the Internet domain and thus we use web ontologies [4].

The main Semantic Web languages are RDF for semantic metadata and OWL for web ontologies. They are introduced in section II. Their relation is analogous to the one between XML for metadata and XML Schema for metadata structuring, although in a semantic, and not only syntactic, information space.

We will use OWL as the tool to formalise ODRL semantics. This formalisation will be accomplished in two phases. First, the lightweight semantics encoded in the ODRL XML Schemas will be translated to OWL ontologies that make them explicit. This is detailed in section III.

Second, it is time for the data dictionary semantics informally written down as textual definitions. It is difficult to formalise them but even if the formalisation is incomplete, they will greatly facilitate ODRL applications development. A preliminary attempt in this direction is shown in section IV.

II. SEMANTIC WEB LANGUAGES OVERVIEW

The Semantic Web paradigm is an attempt to leverage the Web from a distributed information repository to a distributed knowledge one. The Semantic Web basic tools are the Resource Description Framework (RDF) [5] and RDF Schema [6]. A more advanced tool is the Web Ontology Language (OWL) [7].

RDF is used to associate metadata to resources in order to make information about them explicit. Resources are named using URIs, i.e. URLs or URNs. The RDF modelling primitive is the graph. It is composed by a set of arcs used to assert property values about resources and to relate resources between them. Arcs are also called triples in RDF terminology. Each graph arc is composed by a subject URI (the resource about which the statement is made), a property URI and a value (literal) or object URI (the resource to which the subject is related by the property). An RDF description is composed by a set of arcs describing some resources. The set of arcs constitutes a graph that can be navigated in order to retrieve the desired metadata.

As it has been seen until now, RDF provides a framework to model metadata. The basic primitive is the graph. This can be compared with the XML context, where the modelling tool is the tree. However, as an XML tree, an RDF graph is on its own basically unrestricted. Therefore, in order to capture the semantics of a particular domain, some primitives to build concrete "how things are connected" restrictions are necessary.

The tool that provides these restriction-building primitives is RDF Schema. It can be compared to XML Schema or DTDs, which provide building blocks to define restrictions about how XML elements and attributes are related. The primitives are some restricted URI names defined in the RDF and RDFS namespaces. RDFS provides Object Orientation-like primitives. With these primitives, class hierarchies can be defined. Resources are declared members of some of these classes and inherit their associated restrictions.

Moreover, there is a special kind of class: Property. It contains all the resources used to relate subject and object in triples, i.e. all the resources used to name the graph arcs. Property hierarchies can also be defined, and domain (origin) and range (destination) of the RDF graph arcs can be restricted to specific classes.

The Web Ontology Language (OWL) is a more advanced ontology-building toolkit. It provides more fine-grained primitives that allow additional restrictions. OWL is superset of RDF/S, i.e. in an OWL ontology all the primitives of RDF/S can be used.

III. MAKING ODRL XML SCHEMAS SEMANTICS EXPLICIT

As we have said, XML Schemas define the ODRL language syntax but also some simple semantics. The substitution group relations among elements and the extension/restriction base ones among complex types encode generalisation hierarchies.

There are many attempts to make XML metadata semantics explicit, usually they translate it to Semantic Web languages that facilitate the formalisation. Some of them just model the XML tree using the RDF primitives [8]. Others concentrate on modelling the knowledge implicit in XML languages definitions, i.e. DTDs or the XML Schemas, using web ontology languages [9], [10], [11]. Finally, there are attempts to encode XML semantics integrating RDF into XML documents [12], [13].

However, none of them facilitates an extensive transfer of XML metadata to the Semantic Web in a general and transparent way. Their main problem is that the XML Schema implicit semantics are not made explicit when XML metadata instantiating this schemas is mapped. Therefore, they do not take profit from the XML semantics and produce RDF metadata almost as semantics-blind as the original XML. Alternatively, they capture this semantics but they use additional ad-hoc semantic constructs that produce less transparent metadata.

Therefore, we have chosen the ReDeFer methodology [14] that combines a XML Schema to web ontology mapping, called XSD2OWL, with a transparent mapping from XML to RDF, XML2RDF. The ontologies generated by XSD2OWL are used during the XML to RDF mapping in order to generate semantic metadata that makes XML Schema semantics explicit. Both steps are detailed next and then their application to ODRL is shown.

A. *XSD2OWL Mapping*

The XML Schema to OWL mapping is responsible for capturing the schema implicit semantics. This semantics are determined by the combination of XML Schema constructs. The XSD2OWL mapping is based on translating this constructs to the OWL ones that best capture their semantics. These translations are shown in Table 2.

The XSD2OWL mapping is quite transparent and captures a great part of XML Schema semantics. The same names used for XML constructs are used for OWL ones, although in the new namespace defined for the ontology. Therefore, it produces OWL ontologies that make explicit the semantics of the corresponding XML Schemas. The only caveats are the implicit order conveyed by xsd:sequence and the exclusivity of xsd:choice.

Table 2. XSD2OWL translations for the XML Schema constructs and shared semantics with OWL constructs

XML Schema	OWL	Shared informal semantics
element|attribute	owl:DatatypeProperty|ObjectProperty	Named relation between nodes or nodes and values
element @substitutionGroup	rdfs:subPropertyOf	Relation can appear in place of a more general one
element@type	rdfs:range	The relation range kind
complexType|group|attributeGroup	owl:Class	Relations and contextual restrictions package
complexType//element	owl:Restriction	Contextualised restriction of a relation
extension|restriction @base	rdfs:subClassOf	Package concretises the base package
@maxOccurs|@minOccurs	owl:maxCardinality|minCardinality	Restrict the number of occurrences of a relation
sequence|choice	owl:intersectionOf|unionOf	Combination of relations in a context

For the first problem, owl:intersectionOf does not retain its operands order. There is no clear solution that retains the great level of transparency that has been achieved. The use of RDF Lists might impose order but introduces ad-hoc constructs not present in the original metadata. Moreover, as it has been demonstrated in practise, the elements ordering does not contribute much from a semantic point of view. For the second problem, owl:unionOf is an inclusive union, the solution is to use the disjointness OWL construct, owl:disjointWith, between all union operands in order to make it exclusive.

B. *XML2RDF Mapping*

Once all the metadata XML Schemas are available as OWL ontologies, it is time to map the XML metadata that instantiates them. The intention is to produce RDF metadata as transparently as possible. Therefore, a structure-mapping approach has been selected [15]. It is also possible to take a model-mapping approach [16]. XML model-mapping is based on representing the XML information set using semantic tools. This approach is better when XML metadata is semantically exploited for concrete purposes. However, when the objective is semantic metadata that can be easily integrated, it is better to take a more transparent approach.

Transparency is achieved in structure-mapping models because they only try to represent the XML metadata structure, i.e. a tree, using RDF. The RDF model is based on the graph so it is easy to model a tree using it. Moreover, we do not need to worry about the semantics loose produced by structure-mapping. We have formalised the underlying semantics into the corresponding ontologies and we will attach them to RDF metadata using the instantiation relation rdf:type.

The structure-mapping is based on translating XML metadata instances to RDF ones that instantiate the corresponding construct in OWL. The more basic translation is between relation instances, from xsd:elements and xsd:attributes to rdf:Properties. Concretely, owl:ObjectProperties for node to node relations and owl:DatatypeProperties for node to values relations. Values are kept during the translation as simple types and RDF blank nodes are introduced in the RDF model in order to serve as source and destination for properties. They will remain blank until they are enriched with semantic information. For the moment, the current state of the mapping is shown in Fig. 1.

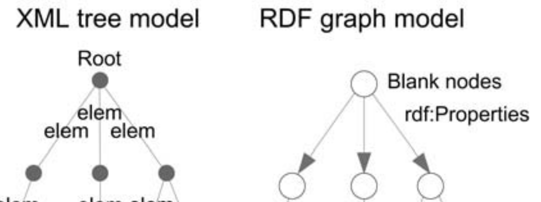

Fig. 1. XML tree and resulting RDF graph models

The current RDF graph model contains all that we can obtain from the XML tree. It is already semantically enriched thanks to the rdf:type relation that connects each RDF property to the owl:ObjectProperty or owl:DatatypeProperty it instantiates. It can be enriched further if the blank nodes are related to the owl:Class that defines the package of properties and associated restrictions they contain, i.e. the XML Schema complexTypes. This semantic decoration of the graph is formalised using rdf:type relations from blank nodes to the corresponding OWL classes.

At this point, we have obtained a semantics-enabled representation of the input metadata. The instantiation relations can now be used to apply OWL semantics to metadata.

C. Application to ODRL XML Schemas

First of all, the XSD2OWL mapping has been applied to the ODRL XML Schemas. ODRL schemas define a quite flat set of hierarchies for complexTypes and elements. They are translated to OWL classes and properties hierarchies as shown in Fig. 2 and Fig. 3 respectively.

Once in OWL form, the previously hidden semantics can be exploited by OWL-aware tools that facilitate implementing ODRL applications.

Applications usually operate over ODRL instances, i.e. XML documents instantiating the XML Schemas. Therefore, in order to take profit from the just formalised semantic, it is necessary to map the XML instances to the semantic enriched form, i.e. to RDF metadata that instantiates the OWL ontologies just created.

The XML2RDF mapping resolves this. It receives the XML metadata for ODRL rights expressions and produces the RDF graph that models the corresponding XML tree. As it has been shown, the RDF graph is enriched with the XML Schema hidden semantics. Now, Semantic Web tools can easily put the ODRL XML Schemas semantics into practice.

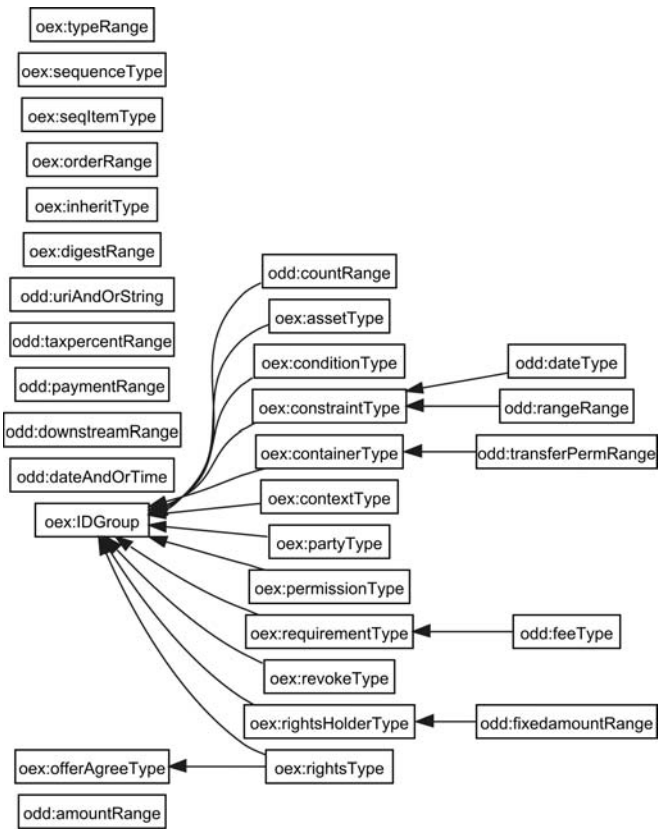

Fig. 2. ODRL XML complexTypes formalised as OWL classes hierarchies. The "Range" suffixed classes correspond to implicit complexTypes

For instance, we will retake the introduction problem about a query for retrieving the constraints affecting a ODRL rights expression. When we are working with the XML version, we need 23 XPath queries in order to retrieve all possible kinds of constraints. However, with the RDF version connected to the ODRL ontologies, a semantic query for oex:constraintElement will be automatically propagated in order to retrieve all the particular constraints defined as substitutionGroups.

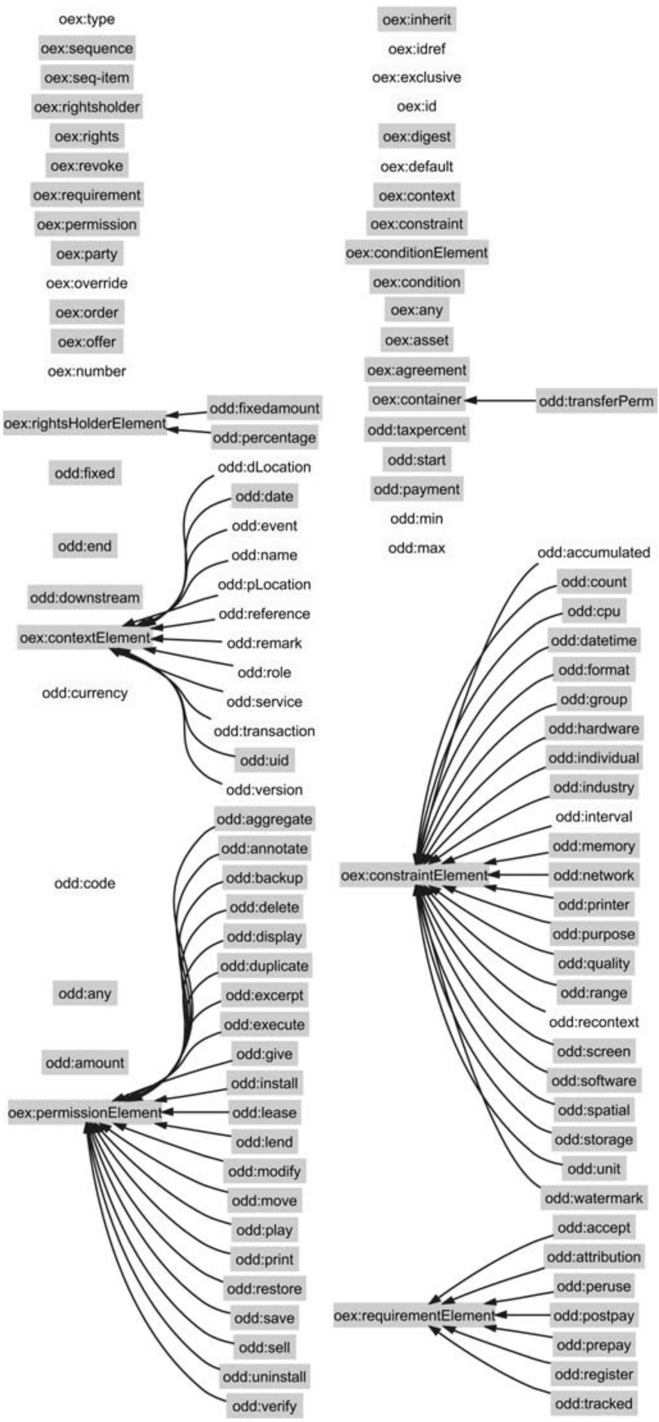

Fig. 3. ODRL XML elements and attributes formalised as OWL properties hierarchies. Grey properties correspond to object properties and white ones to datatype properties

D. *Mapping results*

As a result of the first step of ODRL semantics formalisation shown in this section, we have a methodology and some tools that allow us translating XML ODRL rights expressions into RDF-OWL.

The ODRL OWL ontologies formalise the XML Schema implicit semantics so they are available for Semantic Web tools in order to facilitate ODRL applications implementation. The ODRL Ontologies and metadata examples related to this section are available at [17].

Moreover, the ontologies will serve as the anchor point where more detailed semantics will be attached during the second step of ODRL semantics formalisation. This process is detailed in the next section.

IV. ODRL FORMALISATION USING AN IPR ONTOLOGY

The first step of ODRL semantics formalisation provides the lightweight semantics implicit in ODRL XML Schemas. Moreover, it provides the anchor points where we are going to attach the more detailed semantics formalised from the textual definitions of the Data Dictionary. The detailed semantics are written down as text so, in order to automatically extract them we would need natural language processing (NLP) methods. However, NLP techniques are not advanced enough to fully extract the intended semantics from the short descriptions of the Data Dictionary.

We use a different approach. An accurate reading of the definitions together with the whole ODRL specification will be done, i.e. automatic means are not used. This reading is intended for interpreting ODRL semantics in the framework of an Intellectual Property Rights Ontology, IPROnto [18, 19].

IPROnto is also a OWL web ontology that provides a general semantic framework for the Intellectual Property Rights (IPR) domain. IPROnto is presented in section IV.A. IPROnto guides the formalisation of ODRL semantics. The ODRL ontologies are connected to IPROnto following the interpretation of the ODRL specification. These mappings are detailed in section IV.B and IV.C. Finally, the benefits of the IPROnto-assisted formalisation of ODRL semantics are presented in section IV.D.

A. *IPROnto*

IPROnto is an ontology that tries to formalise the IPR domain from a general and purpose independent point of view. The ontology covers more than just the end user part of the intellectual property value chain. IPROnto models the full value chain and thus it must consider also the intellectual property rights part and not just the usage one. Moreover, it is not restricted to digital media. Therefore, it considers the general creation concept in detail as it is shown next.

IPROnto is firstly based on Intellectual Property literature and regulations, mainly from the World Intellectual Property Organisation (WIPO, http://www.wipo.org). The different IP aspects of IPROnto are detailed in the next subsections.

1) *Creation Model*: the core concepts of IPROnto are those that formalise the notion of creation. As we can see in Fig. 4, there are three points of view of a creation: the abstraction, manifestation and expression perspectives.

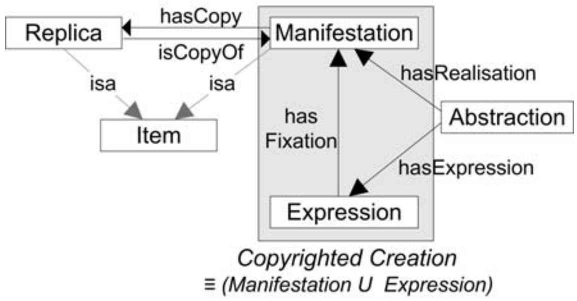

Fig. 4. Creation Model

For instance, if we take the creation “Les Misérables”, we can observe it from these three perspectives taking different forms. From the manifestation view, we can see a script, a book, etc. Its film projection would be seen from the expression perspective. All have in common the original Victor Hugo’s idea visible from the abstraction perspective. The ideas cannot be copyrighted so they lay outside the copyrighted creation concept. Abstraction, on the other hand, is what we grasp as common in different manifestations, expressions or replicas and what allows us saying that they are the same creation.

2) *Rights Model*: from the legal point of view, WIPO recommendations have been followed and the intellectual property rights they define are present in IPROnto. Table 3 shows the included rights hierarchy starting from Copyright. There are also other intellectual property rights that are not shown, e.g. sui-generis rights, neighbor rights, etc. although they are unimportant in this context.

Table 3. Copyright hierarchy

Copyright			
	MoralRight		
		DisseminationRight	
		PaternityRight	
		RespectRight	
		WithdrawalRight	
	ExploitationRight		
		TransformationRight	
			AdaptationRight
			TranslationRight
			SubtitlingRight
		CommunicationRight	
			BroadcastRight
			PublicPerformanceRight
		DistributionRight	
			RentalRight
		ReproductionRight	
			FixationRight

The more important rights in the Digital Rights Management context are Exploitation Rights as they are related to productive and commercial aspects of intellectual property. Each of these rights defines a set of actions that can be done or not on a creation depending on the rights situation:

- *Transformation Right*: grants actions of type transform that produce a new creation, like adapt, translate, subtitle, etc.
- *Communication Right*: grants actions of type communicate, like broadcast, perform, make available (e.g. on the Internet), etc.
- *Distribution Right*: grants actions of type distribute, like sell, rent, etc. This right, and consequently the kind of actions it includes, only affects manifestations of a creation (e.g. compact disk, DVD, cassette, etc.).
- *Reproduction Right*: grants actions of type reproduce, like copy, fix (an expression into a manifestation, e.g. an opera into a CD), etc.

Moral rights are always hold by the creator and cannot be commercially exploited. Moreover, they are only fully considered in Continental-like IPR systems, i.e. legal system like those in the European Union. On the other hand, legal systems of the Anglo-Saxon kind do not consider them. Therefore, as they do not have commercial interest, moral rights are modelled but not detailed in IPROnto for the moment.

We can also identify two more kinds of actions that are related to intellectual property, although the mentioned rights do not cover them:

- *Transfer*: these are actions to move rights between rights holders and are related to the exploitation aspect of intellectual property rights, only exploitation rights can be transferred. End users do not hold rights so there are no transfers to them. There are also commercial actions, which are related to transfer actions. Commercial actions are offer, agree, counteroffer, post-agree, etc.
- *Use*: end users do not hold exploitation rights. They just consume creations, i.e. they use them. Uses are not covered by copyright. However, this does not mean that end users can do whatever they want, they should not realise actions that require copyright. Moreover, they might be subject to special conditions under which they have acquired the permission to use a creation (e.g. a film that can only be viewed a fixed number of times and thus is cheaper than a DVD reproduction).

The previous actions are associated to the different roles that take part in the creation' life cycle. Or, from the commercial point of view, it can be seen as the creation's value chain. Legal persons play these roles. Actions are shown as arrows in Fig. 5. The ovals represent the different roles; those at the source of the arrows perform the actions. The arrow destinations show the role that receives the responsibility over the creation once the action has been performed.

First of all, the creator acts and a new creation is produced. Automatically, there is a holder that gets rights on the creation. The ovals represent roles that might be played by the same person. Therefore, the rights holder can be the same person that acted as creator.

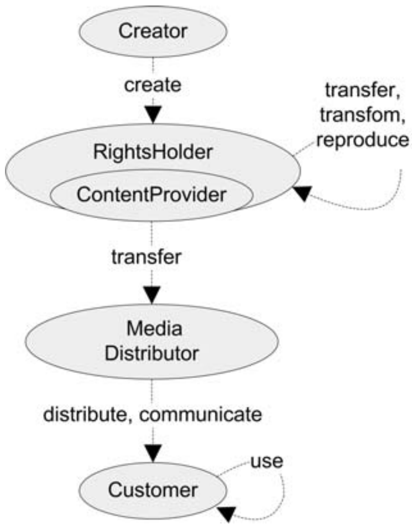

Fig. 5. Creation life cycle through the hands of the different roles involved and the actions they perform to move the creation forward

Then, the rights holder can transfer all or a portion of the rights to a content provider. Content providers are specialised in transforming raw creations in order to facilitate their commercialisation. Moreover, if the creation is commercialised physically, they are responsible for reproducing the creation in order to produce the replicas for consumption.

Next, it is time to make the creation available to end-users. Media distributors are responsible for this part. The get a transfer of the rights they need for the distribute and communicate actions, which are the actions that make creations available for end users.

Finally, at the end of the life cycle or value chain, the costumer uses the creation in order to consume it.

3) *IPROnto in "action"*: as it has been shown, IPROnto takes IP rights into account but it has actions as its central building block, where actions are those covered by exploitation rights but also usage and transfer ones. With them, we try to cover all the events in the value chain.

Actions are not isolated entities, they are related to a bunch of entities that take part or are affected by the action. Moreover, there are space-time coordinates that situate the action. One thing that all actions have in common is that they are verbs. Therefore, in order to facilitate their modelling, we have incorporated into IPROnto ideas from the linguistics field related to the classification of verbs and their relation to other linguistic components.

These relations are called thematic roles or case roles [20] and are classified into initiator, resource, goal and essence. In Table 4 we show the case roles we have considered in IPROnto and also the kinds of verbs they are related to. These kinds of verbs define verbs facets, not disjoint classes of verbs, and concretise the general thematic roles as shown in each row. Therefore, the same verb can present one or more of these facets. For instance, the play verb can show the action, temporal and spatial facets in a particular sentence.

Table 4. General thematic roles (top row) and their concretisations corresponding to their relation to different verb facets (left column)

	initiator	resource	goal	essence
Action	agent, effector	instrument	result, recipient	patient, theme
Process	agent, origin	matter	result, recipient	patient, theme
Transfer	agent, origin	instrument, medium	experiencer, recipient	theme
Spatial	origin	path	destination	location
Temporal	start	duration	completion	pointInTime
Ambient	reason	manner	aim, consequence	condition

Fig. 6 shows an example of action modelling using thematic roles to relate the verb to its participants and context. In this case it is a reproduction of a master copy to produce CDs. It is done using a computer and is completed in 2000.

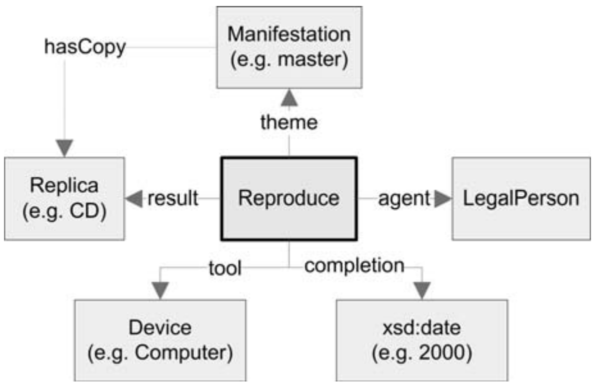

Fig. 6. Action modelling example using thematic roles

To conclude, IPROnto is enriched with general concepts for time, space, tools, part hood, etc. They are taken from upper level ontologies, which define general concepts. We need also specific concepts, e.g. digital media concepts, which are taken from domain ontologies. For instance, we have considered some upper ontologies and domain ontologies:

- Upper ontologies: IEEE SUMO [21], DOLCE [22] and LRI-Core [23]. They define general concepts; in the latter case with a clear legal bias. The other ones are general but include some legal aspects too.
- Domain ontologies: MPEG-7 ontology and TVAnytime ontologies. They are generated automatically from XML Schemas like ORDL ontologies.

B. *Preparing ODRL Ontologies to IPROnto mappings*

First of all, in order to facilitate mappings, some changes are introduced in the ODRL ontologies that were automatically generated from the ODRL XML Schemas. As it is shown in Fig. 2 and Fig. 3, elements are more richly structured than complexTypes. As a consequence, the OWL properties hierarchy is more complex than the OWL classes one.

The common situation for ontologies is the reverse one. Classes use to have richer hierarchical structure than classes and this is the case for IPROnto. Therefore, in order to facilitate mappings, the ODRL classes' hierarchy is enriched. We do not introduce any supplementary knowledge. The objective is simply to replicate the properties hierarchy structure in the classes' hierarchy.

The current lack of structure is because ODRL does not define more specific complexTypes for requirementType, permissionType and constraintType, since they are not needed while working with XML. On the other hand, the corresponding elements (requierementElement, permissionElement and constraintElement) have more specific elements, which appear as their subproperties in the OWL ontology, i.e. play, software, prepay, etc.

Therefore, in order to replicate structure, we introduce a new class for each one of these properties and define the class as a subclass of the corresponding existing class. For instance, the PlayType class is introduced, corresponding to the play property, and it is defined as subclass of permissionType. The same is done for all the subproperties of requierementElement, permissionElement and constraintElement.

The same applies for offer and agree, both related to the offerAgreeType complexType. The corresponding offerType and agreeType are introduced.

As the last preparatory step, we have also reintroduced in the ODRL ontologies all the abstract elements defined in the ODRL specification but not present in the XML Schemas. Consequenly, as detailed previously, we have also introduced the corresponding classes in order to replicate the new properties in the classes' hierarchy. They are use, reuse, transfer and asset management as permissionElement subproperties; interaction, fee and usage as requirementElement subproperties; user, device, bounds, aspect, target, temporal and rights as constraintElement subproperties.

C. *Planning ODRL Ontologies to IPROnto mappings*

Thanks to the previous preparatory step, we have new versions of ODRL ontologies that are easier to relate to IPROnto. We are currently planning the needed mappings in order to effectively produce the integration. It is work in progress so we are going to depict here the principles and techniques we are using. Moreover, we give some mapping examples.

The integration is performed using two techniques. First, for simple cases, it is possible to connect directly ontologies using OWL primitives for concept inclusion and equivalence (e.g. subClassOf, subPropertyOf, equivalentClass, equivalentProperty, sameIndividualAs, etc.).

These are some simple mapping examples (o-ex prefix refers to concepts generated directly from ODRL-EX, o-dd for ODRL-DD, o-ont for the extensions generated during the previous preparatory step and ipro for concepts in IPROnto):

- o-ex:permissionType –subClassOf→ ipro:Verb
- oddo:usageType –subClassOf→ ipro:Use
- oddo:offerType –subClassOf→ ipro:Offer
- oddo:transferType –subClassOf→ ipro:Transfer
- o-dd:individual –subPropertyOf→ ipro:agent
- o-ex:asset –subPropertyOf→ ipro:essence
- o-dd:uid –equivalentProperty→ rdf:ID
- o-dd:name –equivalentProperty→ rdf:label
- etc.

However, the previous technique is only possible when we are mapping one concept from an ontology to one concept in the other ontology. When the conditions for the mapping are more complex, we are using semantic rules [24]. Rules are particularly useful when the mapping must cope with a difference in the manner the concepts are structured in the mapped ontologies.

For instace, the ODRL context element is not used in IPROnto. Web ontologies use the RDF identifier (rdf:ID) instead of the ORDL one (o-dd:uid) and RDF identifiers are directly attached to the concept they identify. In ODRL words this means that the identifier is a direct attribute of the asset. The same applies to the rest of the context model elements.

Therefore, the context element must be removed when mapping an ODRL instance to IPROnto. However, it is easier to convert the context of a contextualised type because it has all this information directly attached, while the contextualised type is empty. For instance, a contextualised description of an offer asset, see Fig. 7, is transformed using the previous simple mappings in conjunction with the mapping rule (1) to the IPROnto-aware description shown in Fig. 8.

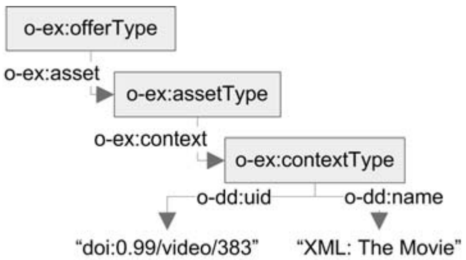

Fig. 7. ODRL example in RDF graph form

$$\text{o-ex:asset}(?x,?y) \wedge \text{o-ex:assetType}(?y) \wedge \text{o-ex:context}(?y,?z) \Rightarrow \text{ipro:Creation}(?z) \wedge \text{o-ex:asset}(?x,?z) \quad (1)$$

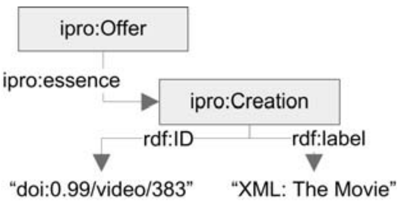

Fig. 8. IPROnto-aware graph resulting from mapping Fig. 7

D. *IPROnto-ODRL benefits*

The direct benefit of the ODRL to IPROnto mappings is that a substantial part of ODRL semantics are formalised. This might reduce ambiguities, or at least highlight possible ambiguous points. Moreover, there are new application development facilities. In addition to the semantic queries benefits shown before, other semantics-enabled tools can be used. One of the most promising tools is Description Logics (DL) [25]. OWL is based on DL so it can be directly fed into DL classifiers. Classifiers are specialised logic reasoners that guarantee computable results. DL classifiers are used with IPROnto in order to automatically check IP uses against the use patterns specified in IP agreements or offers. This facilitates checking if a particular use is allowed in the context of a set of licenses or finding an offer that enables it, once an agreement is reached.

DL classifiers can be directly reused so there is no need to develop ad-hoc applications to perform this function. Moreover, as they are completely OWL semantics aware, the IPROnto to ODRL ontologies mappings enables their use in order to check uses against ODRL licenses, even if they are in XML form. XML ODRL licenses can be mapped to RDF using XML2RDF and then, through mappings, get connected to the IPROnto semantic framework.

The use of DL classifiers for digital rights management, once mapped to IPROnto, can be exemplified with the following scenario:

1) The initial situation is: "USER1 is trying to access a given video stream from a streaming server at 9:30:10 UTC on 2005-04-10". The streaming server implements digital rights management. It inquires the license manager if the current usage is permitted. In order to do that, the streamer models this usage using IPROnto, see Fig. 9, and sends it to the license manager, e.g. as a RDF/XML serialisation.

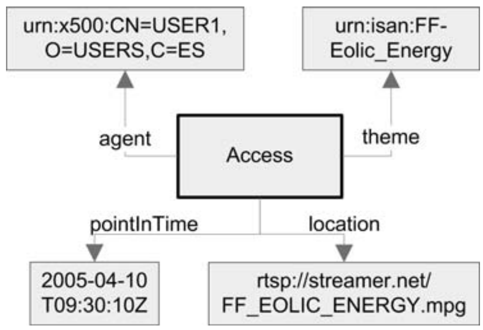

Fig. 9. Usage instance modelled by the streaming server

2) The license manager contains licenses modelled using IPROnto, including the one shown in Fig. 10. This license defines a usage pattern for a creation located at the streaming server that can be performed by a class of agents for a given period of time starting on a given date. Moreover, the license manager has additional metadata stating that USER1 is an instance of the "O=USERS,C=ES" class, which models a group of users.
3) The license manager checks if there is any license that grants a usage pattern that subsumes the usage instance. This can be performed easily and efficiently using a DL

classifier. However, there are some problems that should we resolved before. First, the usage patterns have a condition property that should be ignored during subsumption computation. Second, the usage patterns define time intervals using a start time and duration, while the usage instance defines a time point. In order to check if the time point is included in the time interval, we must use a DL classifier capable of dealing with custom datatypes reasoning [26]. Then, the time interval is translated to a real interval (2) and the time point to a real (3).

$$\text{pointInTime}.\geq[20050401]\ \text{real} \cap \leq[20060401]\ \text{real} \quad (2)$$

$$\text{pointInTime}.=[20050410.093010]\ \text{real} \quad (3)$$

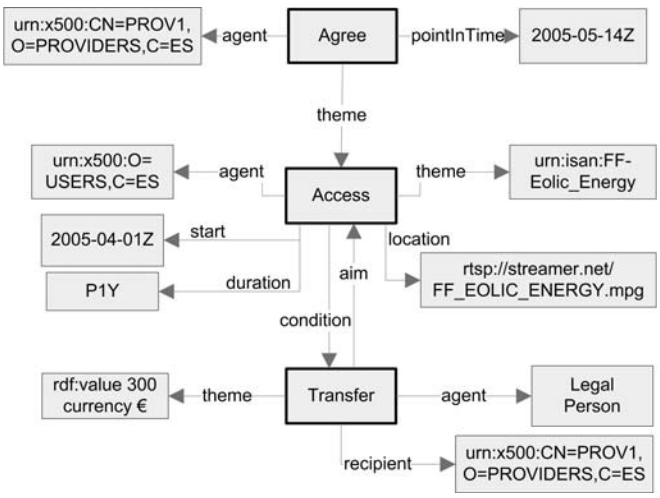

Fig. 10. Use license model defining permitted usage pattern and condition

4) After applying the previous adaptations, subsumption is computed. The usage might be classified in one or more usage patterns. In this case, we test if the usage pattern is the theme of an Agree event. This is equivalent to the agreement authorising this use. Finally, if the usage conditions are satisfied, the license manager tells the streaming server that the use is authorised. Otherwise, it is forbidden.

This is a simple scenario for illustrative purposes. It could be extended in many ways. For instance, if the usage pattern is the theme of an offer, another possibility is to recommend the user the possibility to negotiate it in order to arrive to a new agreement. From this point, this IPR reasoning framework can be connected to negotiation architectures previously developed in our research group [27, 28] in order to achieve assisted negotiation of digital goods.

V. Conclusion

As it has been shown, the Semantic Web approach to ODRL semantics formalisation has started to give its fruits. Even the first step of semantics formalisation, during which the implicit semantics of ODRL XML Schemas have been formalised, has proved very useful simply by making semantic queries possible.

The second step, during which more complex semantics are being defined, is showing promising results and it can greatly enlarge semantic benefits for ODRL applications implementation.

To conclude, it is important to remark that all this work has been done for the current version of ODRL, version 1.1. This version was intended for XML representation and this has made the connection of ODRL ontologies to IPROnto harder. For future versions of ODRL, it might be interesting to consider this possibility, which might enable a more complete formalisation using web ontologies.

References

[1] Iannella, R.: "Open Digital Rights Language (ODRL), Version 1.1". World Wide Web Consortium, W3C Note, 2002. Available: http://www.w3.org/TR/odrl

[2] Delgado, J.; Gallego, I. & García, R.: "Use of Semantic Tools for a Digital Rights Dictionary". In K. Bauknecht, K.; Bichler, M. & Pröll, B. (ed.) "E-Commerce and Web Technologies: 5th International Conference". Springer-Verlag, LNCS Vol. 3182, pp. 338-347, 2004

[3] Berners-Lee, T.; Hendler J. and Lassila O.: "The Semantic Web". Scientific American, May 2001. Available: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

[4] Hendler, J. "Agents on the Semantic Web". IEEE Intelligent Systems, Vol. 16, No. 2, March-April 2001. Available: http://www.ai.mit.edu/people/jimmylin/papers/Hendler01.pdf

[5] Lassila, O. and Swick, R.R. (eds.): "Resource Description Framework (RDF), Model and Syntax Specification". W3C Recommendation, 22 February 1999. Available: http://www.w3.org/TR/REC-rdf-syntax

[6] Brickley, D. and Guha, R.V. (eds.): "RDF Vocabulary Description Language 1.0: RDF Schema". W3C Working Draft, RDF Core Working Group, 2002. Available: http://www.w3.org/TR/rdf-schema

[7] Dean, M. and Schreiber, G. (eds.): "OWL Web Ontology Language Reference". W3C Candidate Recommendation, Web Ontology Working Group, 2003. Available: http://www.w3.org/TR/owl-ref

[8] Klein, M.C.A.: "Interpreting XML Documents via an RDF Schema Ontology". "Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)". pp. 889-894, 2002

[9] Amann, B.; Beeri, C.; Fundulaki, I. & Scholl, M.: "Ontology-Based Integration of XML Web Resources". "Proceedings of the 1st International Semantic Web Conference (ISWC 2002)". pp. 117-131, 2002

[10] Cruz, I.; Xiao, H. & Hsu, F.: "An Ontology-based Framework for XML Semantic Integration". "Eighth International Database Engineering and Applications Symposium". *Coimbra, Portugal*, 2004

[11] Halevy, A.Y.; Ives, Z.G.; Mork, P. & Tatarinov, I.: "Piazza: Data Management Infrastructure for Semantic Web Applications". "12th International World Wide Web Conference". 2003

[12] Lakshmanan, L. & Sadri, F.: "Interoperability on XML Data". "Proceedings of the 2nd International Semantic Web Conference (ICSW 03)". 2003

[13] Patel-Schneider, P.F. & Simeon, J.: "The Yin/Yang web: XML syntax and RDF semantics". "Proceedings of the 11th International World Wide Web Conference (WWW2002)". pp. 443-453, 2002

[14] ReDeFer, http://rhizomik.upf.edu/redefer

[15] Klein, M.C.A.: "Interpreting XML Documents via an RDF Schema Ontology". "Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)". pp. 889-894, 2002

[16] Tous, R. & Delgado, J.: "Using OWL for Querying an XML/RDF Syntax". Poster at WWW'05

[17] ODRL Ontologies, http://dmag.upf.edu/ontologies/odrlontos

[18] Delgado, J.; Gallego, I.; Llorente, S. & García, R.: "Regulatory Ontologies: An Intellectual Property Rights approach". In Meersman, R. & Tari, Z. (ed.): "On The Move to Meaningful Internet Systems 2003:

OTM 2003 Workshops". *Springer-Verlag*, Vol. 2889, pp. 621-634, 2003. ISBN: 3-540-20494-6

[19] Delgado, J.; Gallego, I.; Llorente, S. & García, R.: "IPROnto: An Ontology for Digital Rights Management". In Bourcier, D. (ed.): "Legal Knowledge and Information Systems". *IOS Press, Amsterdam,* Frontiers in Artificial Intelligence and Applications Vol. 106, 2003

[20] Sowa, J.F.: "Knowledge Representation. Logical, philosophical and computational foundations". *Brooks Cole Publishing Co.*, 2000

[21] Pease, A.; Niles, I. & Li, J.: "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications". In Pease, A. (ed.): "Ontologies and the Semantic Web, Papers from the AAAI Workshop". *AAAI Press*, 2002

[22] Gangemi, A.; Guarino, N.; Masolo, C.; Oltramari, A. & Schneider, L.: "Sweetening Ontologies with DOLCE". In Gómez-Pérez, A. & Benjamins, V. (ed.): "Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002". *Springer-Verlag,* pp. 166-181, 2002

[23] Breuker, J.: "Constructing a legal core ontology: LRI-Core". In Proceeding of "Workshop on Ontologies and their Applications". *Sao Luis, Maranhao, Brazil*, 2004

[24] Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosof, B. & Dean, M.: "SWRL: A Semantic Web Rule Language Combining OWL and RuleML". *DARPA DAML Program,* Draft, No. 0.7, 2004. Available: http://www.daml.org/rules/proposal

[25] Pan, J.Z.: "Description Logics: "Reasoning Support for the Semantic Web". School of Computer Science, The University of Manchester, Oxford Rd, Manchester M13 9PL, UK, 2004

[26] Pan, J..Z. & Horrocks, I.: "OWL-Eu: Adding Customised Datatypes into OWL". In "Proc. of Second European Semantic Web Conference (ESWC 2005)". *Springer-Verlag*, 2005

[27] Delgado, J.; Gallego, I.; García, R. & Gil, R.: "An architecture for negotiation with mobile agents". In A. Karmouch; T. Magedanz & J. Delgado (ed.) "Mobile Agents for Telecommunication Applications: 4th International Workshop". *Springer-Verlag*, pp. 21-31, 2002

[28] Gil, R.; García, R. & Delgado, J.: "Delivery context negotiated by mobile agents using CC/PP". In "Mobile Agents for Telecom Applications, (MATA'03)". *Springer-Verlag*, pp. 99-110, 2003