

# An OWL Copyright Ontology for Semantic Digital Rights Management

Roberto García and Rosa Gil

Universitat de Lleida  
Jaume II 69, E-25001 Lleida, Spain  
roberto@griho.net  
rgil@diei.udl.es

**Abstract.** Digitalisation and the Internet have caused a content reproduction and distribution revolution with clear implications for copyright management. There are many Digital Rights Management (DRM) efforts that facilitate copyright management in closed domains but they find great difficulties when they are forced to interoperate in an open domain like the World Wide Web. In order to facilitate interoperation and automation, DRM systems can be enriched with domain formalisations like the Copyright Ontology. This ontology is implemented using the Description Logic variant of the Web Ontology Language (OWL-DL). This approach facilitates the implementation of efficient usages against licenses checking, which is reduced to description logics classification.

## 1 Introduction

Recently, there have been great changes in the copyright market motivated by the digital and Internet revolutions. First, these revolutions have introduced new risks in the classical market, which was basically based on the distribution of physical instances of content. Second, they have opened opportunities to create new markets based on digital creations and the Internet distribution medium.

In order to manage this new situation, the main approach is to take profit from the new technological opportunities in order to develop systems to manage and protect digital works. This is referred to as Digital Rights Management, or DRM. A DRM system (DRMS) is build from IT components and services along with the corresponding law, policies and business models.

Due to the globalisation of the digital content market, different DRMSs are being forced to interoperate. One of the main initiatives for DRM interoperability is the ISO/IEC MPEG-21 [1] standardisation effort. The main interoperability facilitation component is the Rights Expression Language (REL), which is based on a XML grammar. Therefore, it is syntax-based. There is also the MPEG-21 Rights Data Dictionary (RDD) that captures the semantics of the terms employed in the REL [2]. However, it does so without defining a formal semantics [3].

The limitations of a purely syntactic approach and the lack of formal semantics can be overcome using a formal semantics approach based on ontologies. They have been used, for instance, to validate and correct inconsistencies in MPEG-21 RDD [4,5].

Another MPEG-21 RDD formalisation is OREL [6]. In any case, these initiatives focus on the RDD semantics, which are too specific to facilitate interoperability.

To the best of our knowledge, there is just another ontological framework for DRM, OntologyX [7]. However, it is a commercial product for which there is little publicly available information. Moreover, OntologyX concentrates on the kind of actions that can be performed on governed content; it does not take into account the underlying legal framework.

In order to build a generic ontological framework that facilitates interoperability, the focus must be placed on the underlying legal, commercial and technical copyright aspects. This is the approach for the Copyright Ontology [8], detailed in Section 2, which produces a general conceptualisation. Therefore, it can be used as an interoperability facilitator for the main DRM standards like MPEG-21 or the Open Digital Rights Language (ODRL) [9].

Moreover, the ontology is implemented as an OWL Web ontology based on the Description Logic (DL) variant, i.e. OWL-DL. This implementation facilitates DRMS development as license checking is implemented using a DL reasoner. The OWL-DL implementation is detailed in Section 3. Finally, the conclusions and the future work are presented in Section 4.

## 2 Copyright Ontology

The copyright domain is a very complex one and conceptualising it is a very challenging task. In order to facilitate this, the Copyright Ontology conceptualisation task has been divided in three parts. Each part concentrates on a portion of the problem. However, each part is not independent from the rest; there are many interrelations among them.

The conceptualisation starts from building a model for the more primitive part, the Creation Model. This model is the basis for building the conceptual models of the rest of the parts. The following step is to build the Rights Model, and then the Action Model is built on the roots of the two previous ones. This section presents the main points of these three models. There are more details in [10].

The Creation Model defines the different forms a creation can take, which are classified depending on three points of view:

- **Abstract:** Work.
- **Object:** Manifestation, Fixation and Instance.
- **Process:** Performance and Communication.

The Rights Model follows the World Intellectual Property Organisation (WIPO) [11] recommendations in order to define the rights hierarchy. It includes economic plus moral rights, as promoted by WIPO, and copyright related rights. The most relevant rights in the DRM context are economic rights as they are related to productive and commercial aspects of copyright. Reproduction, Distribution, Public Performance, Fixation, Communication and Transformation Right are economic rights.

Finally, the Action Model corresponds to the primitive actions that can be performed on the concepts defined in the Creation Model and which are regulated by the rights in the Right Model. For the economic rights, these are the governed actions:

- **Reproduction Right:** *reproduce*, commonly speaking *copy*.
- **Distribution Right:** *distribute*. More specifically *sell*, *rent* and *lend*.
- **Public Performance Right:** *perform*; it is regulated by copyright when it is a public performance and not a private one.
- **Fixation Right:** *fix*, or *record*.
- **Communication Right:** *communicate* when the subject is an object or *retransmit* when communicating a performance or previous communication, e.g. a re-broadcast. Other related actions, which depend on the intended audience, are *broadcast* or *make available*.
- **Transformation Right:** *derive*. Some specialisations are *adapt* or *translate*.

The action concepts are complemented with a set of relations that link them to the action participants. This set is adopted from the linguistics field and it is based on case roles [12,13].

Table 1, it is shown on the top the generic case roles and on the right 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 1.** Case roles

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

The previously introduced pool of primitive actions can be combined in order to build different value chains in the copyright domain. Each of the value chains steps can be detailed using the specific action concept and the corresponding case roles.

In order to illustrate one of the numerous possibilities, Fig. 1 shows a model for all copy actions in a Peer to Peer network performed by agent “granted” who copies “content01” from “PeerA” to two peers from the set “PeerB, PeerC, PeerD” at any time point six months after “2006-01-01”.

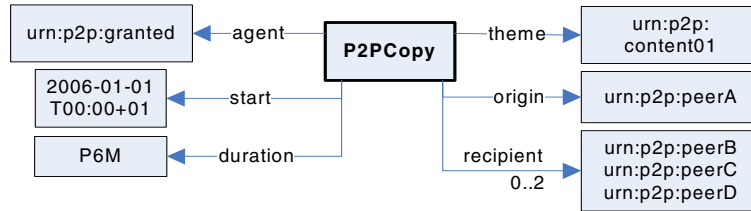


Fig. 1. Model for a copy action in the P2P diffusion scenario

This kind of action patterns can be used to model licenses. There are two additional licensing actions: *Agree* and *Disagree*. They are the building block of any license. Fig. 2 shows a license for the *Copy* action previously shown in Fig. 1. As it is shown, the *condition* case role is used in order to introduce a compensation for the agent that grants the copy action, a 3€ transfer from the granted agent.

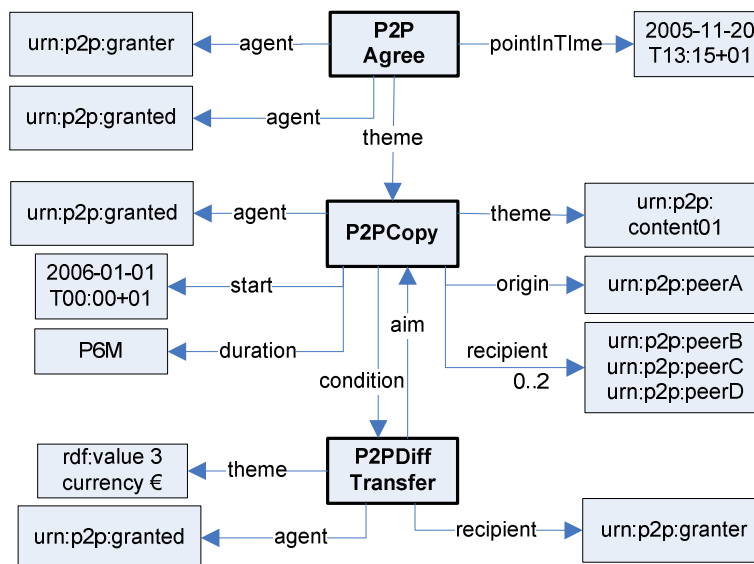


Fig. 2. Model for an agreement on a copy action pattern plus a condition

As it is going to be detailed in the next section, deontic operators are implicit in the agreement model. The agreement *theme* corresponds to an implicit permission, i.e. the theme of an agreement is permitted. The *condition* on the agreement theme correspond to an obligation, i.e. in order to fulfil the theme action it is necessary to satisfy the pattern defined by the condition property object. Finally, it is also possible to model prohibitions. This can be done in two ways, by agreeing on a negated pattern or by using the *Disagree* action.

### 3 OWL Implementation

The main objective has been to provide a straightforward and efficient implementation geared towards an extensive use of DL (Description Logic) reasoners. They are used to automatically check copyright-governed events against the authorised action patterns specified in the licenses. These patterns, the *theme* of an *Agree* or a *Disagree*, are modelled as classes. This facilitates checking if a particular action, once modelled as an instance, is allowed or not.

DL classifiers can be directly reused so there is no need to develop ad-hoc applications to perform this function. The more complex behaviours that cannot be captured using OWL-DL can be modelled using Semantic Web rules.

Licenses are modelled as OWL Classes and copyrighted content intended uses are modelled as instances. Then, in order to check if a usage (instance) is authorised by a set of licenses (classes) a DL reasoner is used to classify the instance in the available classes. If the instance is classified into a class that models an agreement, the *Agree* class as specified in the Copyright Ontology, the usage is authorised.

In order to model the licenses usage patterns, OWL classes and the “owl:Restriction” primitive are used. The restrictions are modelled as necessary and sufficient conditions in order to “trigger” the classification as authorisation mechanism. Each pattern property that is restricted is associated to the corresponding “owl:Restriction” through its “owl:onProperty” relation.

The restricted values for the property are specified with “owl:allValuesFrom” or “owl:someValuesFrom” when the range is a class and “owl:hasValue” when the range is limited to a specific instance. Therefore, it is possible to build patterns that define a concrete instance as the range of one of its case roles, e.g. a concrete user or a specific location. Therefore, very concrete events can also be modelled as classes and used afterwards as patterns or models for the actual action a user is trying to perform.

#### 3.1.1 OWL-DL Implementation of a License Action Pattern

Suppose we want to implement the action pattern in Fig. 1. Table 2 shows the corresponding OWL class. The pattern is for *Copy* actions, so it is a subclass of *Copy*, and it is equivalent to the class resulting from the intersection of four OWL restrictions, which constitute the necessary and sufficient conditions that would trigger the classification of authorised usage instances.

**Table 2.** Class pattern for the actions authorised by the example license

|                       |   |     |
|-----------------------|---|-----|
| Pattern $\sqsubseteq$ | Copy  | (1) |
| Pattern $\equiv$      | $\forall \text{pointInTime} . \geq 2006-01-01T00:00:00, \leq 2006-06-30T23:59:59 \sqcap$  | (2) |
|                       | $\exists \text{agent} . \{ \text{granted} \} \sqcap \exists \text{origin} . \{ \text{peerA} \} \sqcap \exists \text{theme} . \{ \text{content01} \} \sqcap$ | (3) |
|                       | $( \leq 2 \text{ recipient} ) \sqcap$   | (4) |
|                       | $\forall \text{recipient} . \{ \text{peerC}, \text{peerD}, \text{peerB} \}$   | (5) |

### 3.2 Overcoming Open World Assumption

The main problem of the OWL-DL implementation presented in the previous section is that of the Open World Assumption (OWA). This problem arises when the DL reasoners is trying to classify a given usage instance into the existing classes for usage patterns. It can be said that the reasoner is very “conservative” as, although the necessary and sufficient conditions are met, it will not classify an instance into a class if new facts can make it retract from this decision.

In other words, OWL-DL reasoners follow the open world assumption. In some cases this is the desired behavior but this is not the case for this license checking implementation. The intention is to make a local close world assumption and make a decision on the currently available facts as the outcome is to decide if the current action should be authorized or not just right now.

There are some OWL-DL restriction primitives that lead to OWA problems:

- **maxCardinality** ( $\leq n$ ): the reasoner is conservative with this restriction as, although the cardinality restriction might be satisfied at a given time, new facts can make the cardinality greater than  $n$ , i.e. ( $> n$ ). The cardinality restriction is also affected as it is the conjunction of a maxCardinality with a minCardinality.
- **allValuesFrom** ( $\forall R.C$ ): the situation in this case is that, although at the current time all the values for the  $R$  property are in the  $C$  class, in the future, there might be new facts that involve  $R$  with a value not in  $C$ , i.e.  $R.(¬C)$ .

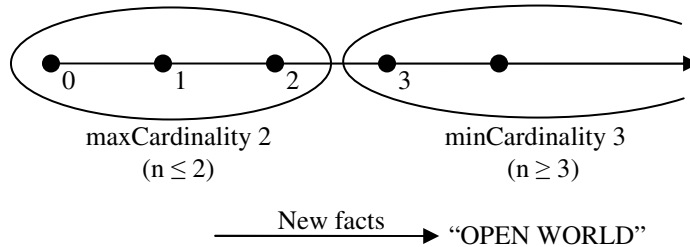
On the other hand, other OWL-DL restrictions, or their combination, are not affected by the OWA and thus do not affect the license checking implementation. Some of them are:

- **minCardinality** ( $\geq n$ ): there is no OWA problem here as once the reasoner can check that the cardinality is equal or greater than  $n$ , i.e. ( $\geq n$ ), new facts cannot make this inference false, i.e. ( $< n$ ).
- **someValuesFrom** ( $\exists R.C$ ): once there is some  $R$  whose value is in  $C$ , new facts cannot make that there does not exist some  $R$  with a  $C$  value, i.e.  $\neg(\exists R.C)$ . Therefore, there is not a OWA problem here.
- **allValuesFrom** ( $\forall R.C$ ) and **FunctionalProperty**  $R$ : the combination of the allValuesFrom, a OWA sensible construct, with the FunctionalProperty in some cases, the combination of a OWA problematic construct with other

There are many ways to overcome OWL-DL’s OWA through epistemic operators [14] and non-monotonic OWL extensions [15]. However, in these cases it is necessary to get outside standard OWL and, what is even more inconvenient, these approaches are basically theoretical and there are not complete implementations of them right now.

In order to implement the OWL-DL based license checker, we have adopted a more pragmatic approach, which does not require additional language constructs neither a reasoner different from the existing OWL-DL enabled ones.

Fig. 3 illustrates this approach. As it can be observed, a maxCardinality restriction defines a set of accepted cardinality values, e.g. from zero to two. However, as new facts are known under an OWA, instances previously classified into this restriction can “get out” of the corresponding set.



**Fig. 3.** Oppose OWA-sensible maxCardinality to OWA-insensible minCardinality

On the other hand, it can be observed that a maxCardinality restriction ( $\leq n$ ) has an opposite set corresponding to the minCardinality restriction ( $\geq n+1$ ). As it has been shown, the minCardinality construct is not affected by OWA. Therefore, the idea is to take profit from this fact and make the reasoner look for the opposite set, the OWA-insensible one, and check that it is not satisfied. This implies that the reverse is satisfied by the current set of facts at hand, overcoming the OWA assumption that makes the reasoner not able to infer that.

Therefore, the approach is to negate the restriction and to undo this at the metalevel, it is to check outside the DL reasoner that the negated restriction is not satisfied and thus it can be inferred that the original one does. The negation is modelled at the metalevel introducing the *Disagree* class, which is the opposite of the *Agree* class.

Therefore, the allValuesFrom restriction on the *theme* class of an *Agree*, i.e.  $\forall R.C$ , is converted into a someValueFrom restriction on the *theme* class of a *Disagree*, i.e.  $\exists R.\neg C$ . On the other hand, the maxCardinality restriction on the *theme* class of an *Agree*, i.e.  $(\leq n R)$ , is converted into a minCardinality restriction on the *theme* class of a *Disagree*, i.e. **Disagree theme** ( $\geq n+1 R$ ).

To summarise, all the OWA-sensible constructs are moved to a new class pattern which is disagreed, i.e. unauthorised, in order to model the metalevel negation. This new class results from the disjunction of all the transformed restrictions and is intersected with the original pattern, which is now composed by just the OWA-insensible restrictions and is what remains as the subject of the *Agree*.

In order to check if a given usage instance is authorised, it is classified in the available license class patterns, both agreed and disagreed ones. If the instance is classified into an agreed class pattern, it is also checked if it has not been classified into the corresponding disagreed class pattern for the OWA-sensible restrictions. If it is so, the usage is authorised. Otherwise, it is not classified in the agreed pattern or it is classified in the disagreed pattern, the usage is not authorised. The following example illustrates this mechanism.

### 3.2.1 OWA-Insensible Implementation Example

Table 3 shows the class patterns that overcome the OWA and result from the previous transformation on the Table 2 OWA-sensible constructs. From lines 4 to 6 there is the *Pattern* that contains the transformed OWL constructs.

Note that *pointInTime* is defined as a functional property in the Copyright Ontology so it is not affected by the OWA and remains unchanged, line 2, like the other OWA-insensible constructs, line 3. All of them in *Pattern'*.

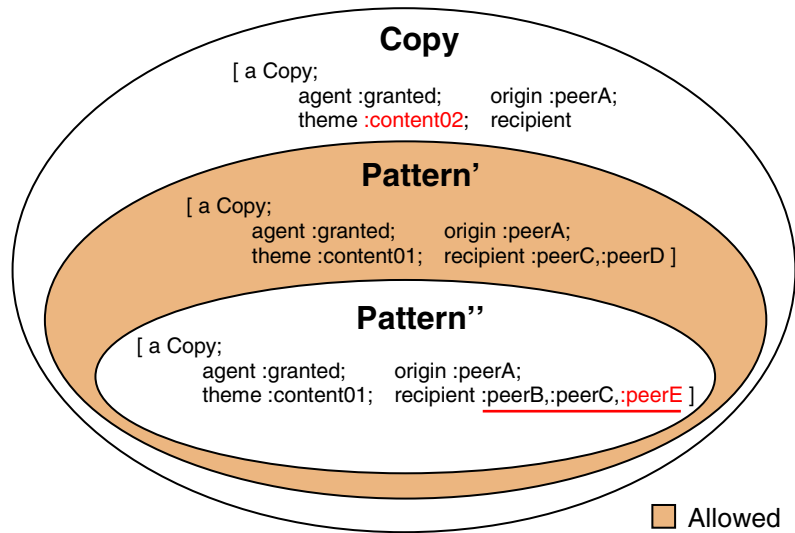
**Table 3.** OWA-aware class patterns for the actions authorised by the example license

|                        |  |     |
|------------------------|--|-----|
| $Pattern' \sqsubseteq$ | Copy   | (1) |
| $Pattern' \equiv$      | $\forall pointInTime. \geq 2006-01-01T00:00:00, \leq 2006-06-30T23:59:59 \sqcap$               | (2) |
|                        | $\exists agent.\{granted\} \sqcap \exists origin.\{peerA\} \sqcap \exists theme.\{content01\}$ | (3) |
| $Pattern'' \equiv$     | $Pattern' \sqcap$  | (4) |
|                        | $( (\geq 3 \text{ recipient}) \sqcup$  | (5) |
|                        | $\exists recipient. (\neg \{peerC, peerD, peerB\}) )$  | (6) |

The overall methodology is exemplified in Fig. 4. It shows three usage instances that are checked against the license patterns presented in Table 3. The first one is classified into *Copy* because the *theme* is “fragment02” and so it cannot be classified into *Pattern'*. Therefore, it is not authorised by the involved license.

The second one is classified into *Pattern'*, but not into *Pattern''*. Consequently, it is in an agreed class pattern and not in the corresponding disagreed pattern so it is authorised.

Finally, the third instance is classified into *Pattern''*, and correspondingly into *Pattern'*, because it satisfies both of its restrictions. There are three or more recipients, i.e. recipient cardinality is  $\geq 3$ , and one of them is not in the authorised set of peers, i.e. there is some recipient which is the set of the unauthorised peer. Therefore, this usage instance is not authorised. Note that, as the transformed patterns



**Fig. 4.** Examples of the license checking methodology to overcome OWA



in *Pattern*'' are disjointed, if any of them is satisfied the usage instance would be unauthorised, as it is the intended behaviour.

### 3.3 Building a Semantic DRMS

The Semantic DRMS [16] puts the Copyright Ontology into practice in order to build a semantics-oriented DRM system. It is implemented at two levels. The ground level is about OWL-DL classes and instances, the ones used to model licenses and usages, and it implements if instances are classified into class patterns. This level can be implemented with a common Description Logic reasoner. Pellet [17] has been selected because it can reason over custom data types and this has been very useful to check licensing time ranges.

Additionally, prior to loading license class patterns into the DL reasoner, it is necessary to perform the required transformations to make the class patterns OWA-insensible. This is done programmatically with Java through the Pellet's OWL API.

However, this is just the implementation at the ground level. All this must be complemented with a metalevel that implements the deontic aspects that are implicit in the conceptual model. The metalevel guides how the DL checks are performed and captures the semantics of the implicit obligations, permissions and prohibitions. Particularly, it implements the metalevel negation that has been employed to overcome the OWA.

The metalevel has been implemented programmatically. However, the most natural option would be to model the metalevel semantics using the Semantic Web Rule Language (SWRL) [18] and implement it using an appropriate rule engine, for instance SweetRules [19].

## 4 Conclusions and Future Work

As it has been shown, the Copyright Ontology constitutes a complete framework for representing copyright value chains and the associated flow of rights situations, agreements, offers, etc. It is build from the key concepts in the copyright domain so it constitutes an ontology for DRM interoperability.

Moreover, it has been implemented using OWL-DL, which has allowed reducing license checking to DL classification. DL reasoning is very efficient but it is based on a Open World Assumption (OWA). License checks are performed at usage time and must take a local Close World Assumption (CWA).

In order to provide that, this paper contributes a generic methodology that makes OWL-DL follow a CWA when required. It is based on negating OWL constructs that lead to an OWA and transforming to the reverse ones, which are not affected by the OWA. This negation is then undone at a metalevel. Altogether, this methodology does not introduce any significant overhead as it just requires an additional DL classification per license check and the transformation is performed at modelling time.

The future work concentrates now in building a complete Semantic DRM system based on the Copyright Ontology and its OWL-DL implementation. Moreover, the objective is to take profit from this generic ontology in order to map MPEG-21 REL expressions to the Copyright Ontology and make the Semantic DRMS MPEG-21 aware.

In order to achieve this, an ontology for MPEG-21 REL has been generated automatically from the XML Schemas that define this standard using the XML Semantics Reuse Methodology [20] and the ReDeFer [21] XSD2OWL tool. These ontologies provide the hook points for integrating MPEG-21 REL expressions into the SemDRMS ontological framework.

## References

1. de Walle, R.V.; Burnett, I.: "The MPEG-21 Book". John Wiley & Sons, UK, 2005
2. Wang, X.; DeMartini, T.; Wragg, B.; Paramasivam, M.; Barlas, C.: "The MPEG-21 rights expression language and rights data dictionary". IEEE Transactions on Multimedia, Vol. 7, No. 3, pp. 408-417, 2005
3. García, R.; Delgado, J.: "An Ontological Approach for the Management of Rights Data Dictionaries". In Moens, M. & Spyns, P. (ed.): "Legal Knowledge and Information Systems". IOS Press, Frontiers in Artificial Intelligence and Applications Vol. 134, 2005
4. García, R.; Delgado, J.; Rodríguez, E.: Ontological Analysis of the MPEG-21 Rights Data Dictionary (RDD). Input document M12495, ISO/IEC JTC1 SC29 WG11 Meeting, Nice, France, 2005
5. Barlas, C.; Rust, G.: Rights Data Dictionary, Technical Corrigendum 1. ISO/IEC JTC1 SC29 WG11, 2005
6. Qu Y.; Zhang, X; Li, H.: OREL: an ontology-based rights expression language. In Proceedings of the 13th international World Wide Web Conference. ACM Press, pp. 324-325, 2004
7. OntologyX, <http://www.ontologyx.com>
8. Copyright Ontology, <http://rhizomik.net/ontologies/copyrightonto>
9. García, R.; Gil, R.; Delgado, J.: "A Web Ontologies Framework for Digital Rights Management". In press, Journal of Artificial Intelligence and Law, Springer, 2006
10. García, R.: "A Semantic Web Approach to Digital Rights Management". PhD Thesis, Technologies Department, Universitat Pompeu Fabra, Barcelona, ES, 2006. <http://rhizomik.net/~roberto/thesis>
11. WIPO, <http://www.wipo.int>
12. Sowa, J.F.: "Knowledge Representation. Logical, philosophical and computational foundations". Brooks Cole Publishing Co., 2000
13. Dick, J.P.: "A conceptual, case-relation representation of text for intelligent retrieval". University of Toronto, Canada, 1991
14. Grimm, S.; Motik, B.: "Closed World Reasoning in the Semantic Web through Epistemic Operators". OWL: Experiences and Direction Workshop, 2005
15. Katz, Y.; Parsia, B.: "Towards a Nonmonotonic Extension to OWL". OWL: Experiences and Direction Workshop, 2005
16. Semantic Digital Rights Management System, <http://rhizomik.net/semdrms>
17. Pellet OWL Reasoner, <http://www.mindswap.org/2003/pellet>
18. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosz, B.; Dean, M.: "SWRL: A Semantic Web Rule Language Combining OWL and RuleML". World Wide Web Consortium Member Submission, 2004
19. SweetRules, <http://sweetrules.projects.semwebcentral.org>
20. García, R.: "Chapter 7: XML Semantics Reuse". In "A Semantic Web Approach to Digital Rights Management". PhD Thesis, Universitat Pompeu Fabra, Barcelona, ES, 2006. <http://rhizomik.net/~roberto/thesis/html/Methodology.html#XMLSemanticsReuse>
21. ReDeFer, <http://rhizomik.net/redefer>