Autores 3.0: Online Intellectual Property Registry using Semantic Technology



Registry Ontology

0.12

Contents

Requirements Specification

The Registry ontology, in comparison with the Basic ontology[1], has been developed to provide more complex answers regarding the intellectual property of a work.

The first step to develop the ontology is to explicitly document the requirements it should satisfy. Ontology requirements specifications will be made using a tabular format [5] showing:

- Purpose: Main goal of the ontology. In other words, the main function or role that the ontology should have.
- Scope: The general coverage and the degree of detail that the ontology should have.
- Implementation Language: The formal language that the ontology should have.
- Intended End-Users: The intended end-users expected for the ontology.
- Intended Uses: The intended uses expected for the ontology.
- Ontology Requirements:
    - Non-Functional Requirements: The general requirements or aspects that the ontology should fulfill, including optionally priorities for each requirement.
    - Functional requirements: The content specific requirements that the ontology should fulfill, in the form of groups of competency questions and their answers, including optionally priorities for each group and for each competency question.
- Pre-Glossary of Terms:
    - Terms from Competency Questions: The list of terms included in the competency questions and their frequencies.
    - Terms from Answers: The list of terms included in the answers and their frequencies.
    - Objects: The list of objects included in the competency questions and in their answers.

The following table formally specifies the requirements of the Registry ontology.

[1] The ontology and its documentation are available in [7].

	Autores 3.0
	Registry Ontology
1	**Purpose**
	This ontology is developed to provide a vocabulary about intellectual property domain agreed by general consent that could be used by users and computer programs to model works and its exploitation rights transmitted by means of licenses and other cession agreements, so it can be known which actions can be legally carried out with a specific work.
2	**Scope**
	The ontology is not only be focused on the intellectual property domain (limited to jurisdictions, licenses, rights and special conditions), but also on works and their classification, their creator and their rights holders. *The granularity level is directly related to the competency questions and the identified concepts.*
3	**Implementation Language**
	The ontology will be implemented in OWL DL due to its enormous power of expressivity.
4	**Intended End-Users**
	User 1. P2P programs to know if certain actions (download, reproduction, interactively available…) regarding a specific work are legal. *User 2. Someone looking for a useful work to carry out actions involving its exploitation.* *User 3. Authors intending to know which kind of licenses do exist.* *User 4. People or authors looking for useful works to integrate them in theirs.* *User 5. Person who needs to know the legal uses that can carry out with a specific work.* *User 6. Person who wants to create his own license.* *User 7. Internet user or webmaster who wants to know the uses of a work he has just downloaded.* *User 8. Computer program who needs to know the intellectual property aspects related to a work.* *User 9. Author looking for a similar one to collaborate with.*

Autores 3.0	
Registry Ontology	
5	**Intended Uses**
	Use 1. P2P program checking if it can allow the download of a work. *Use 2. P2P program checking if it can allow the reproduction of a work or making it available.* *Use 3. License creators who want to register their own license.* *Use 4. Author who wants to know which works are derivative from his works.* *Use 5. Author who registers his own work.* *Use 6. Person checking the legal uses of a specific work.* *Use 7. Automatic discovery of plagiarism or other illegal uses of a work in Internet.* *Use 8. Automatic warning to users about the illegal use of a specific work.* *Use 9. Person looking for a podcast that could be freely distributed.* *Use 10. Musician who is looking for another one that has already published songs with licenses that allow their free reproduction.*
6	**Ontology Requirements**
	a. Non-Functional Requirements
	NFR 1. The ontology will be documented in English and annotated in English and Spanish. *NFR 2. The ontology will be documented in English and Spanish.* *NFR 3. The terminology will be checked on WordNet.*

Autores 3.0	
Registry Ontology	
6	**Ontology Requirements**
	b. Functional Requirements: Groups of Competency Questions
	GCQ 1. Works (and work types) (9 CQ) CQ 1. ¿Has sculpture "Akarys91" been a team work? Yes CQ 2. ¿Which works have been derived from sculpture "Akarys91"? Akarysene and Akarysatos. CQ 3. ¿Which works have sculpture "Akarys91" been derived from? None CQ 4. ¿Which videos have been registered under license CC-BY-SA? Flying with Florian, Meine kleine Tochter im Zoo and A Christmas Dream. CQ 5. ¿Is sculpture "Akarysene" an original work? No CQ 6. ¿Which work types can an author choose when registering his work? Text, Drawing/Plans/Maps, Sculpture, Photography, Choreography, Score, Video, Audio, Software and Database. CQ 7. ¿Which work types have been registered by Silvio Longhi? Audio and Text. CQ 8. ¿Which works can be used with commercial purposes? Flying with Florian, Meine kleine Tochter im Zoo... CQ 9. ¿Which are the titles of the works created by Thomas Schröder? Akarysatos, Meine kleine Tochter im Zoo and La mirada difusa.

Autores 3.0	
Registry Ontology	
6	**Ontology Requirements**
	b. Functional Requirements: Groups of Competency Questions
	GCQ 2. Licenses, rights and special conditions (11 CQ)
	CQ 10. ¿Which work types can be registered under BSD license? Text, Software and Database.
	CQ 11. ¿Which works are registered under personalized licenses? Akarysatos.
	CQ 12. ¿ Which works are registered under Apache license? Akarysene.
	CQ 13. ¿Can a work registered under Apache licensed be publicly performed? Yes
	CQ 14. ¿Which licenses do allow commercial uses? CC–BY, CC–BY–ND, CC–BY–SA
	CQ 15. ¿Which licenses have been used to license text works? CC–BY
	CQ 16. ¿Which license is sculpture "Akarys91" registered under? CC–BY–SA
	CQ 17. ¿Can I use the work "Akarisene" with commercial purposes? Yes
	CQ 18. ¿Which licenses are applicable to photographies? CC–BY, CC–BY–NC, CC–BY–NC–ND, CC–BY–NC–SA, CC–BY–ND, CC–BY–SA
	CQ 19. ¿Which licenses don't force derivative works to be licensed under the license that governs the original work (Share-alike)? CC–BY–NC, CC–BY–NC–ND, CC–BY–ND, MIT.
	CQ 20. ¿Can I create derivative works from "Akarys91"? Yes
	GCQ 3. People (7 CQ)
	CQ 21. ¿Who has informed about the rights of the work "Akarys91"? Antonio Garrido.
	CQ 22. ¿Who owns the rights of the sculpture "Akarys91"? Thomas Schröder
	CQ 23. ¿What is the author's name of the sculpture "Akarys91"? Thomas Schröder
	CQ 24. ¿Who is the creator of license "PL2CL"? Cristina Karlen
	CQ 25. ¿Who have registered sculptures? Thomas Schröder
	CQ 26. ¿Who owns the rights of Antonio Garrido's works? José Garrido
	CQ 27. ¿Which authors have registered works under their own licenses? Antonio Garrido and Cristina Fernández.

Autores 3.0		
Registry Ontology		
6	**Ontology Requirements**	
	b. Functional Requirements: Groups of Competency Questions	
	GCQ 4. Registry and Jurisdictions (4 CQ) CQ 28. ¿In which jurisdiction was the work "Akarys91" created? Canadian Jurisdiction CQ 29. ¿When was the work "Akarys91" registered? 06/July/2009 23:26 GMT+1 CQ 30. ¿Which registered works have been created in the German jurisdiction? Akarys91 and Meine kleine Tochter im Zoo. CQ 31. ¿What is the URL where can be checked the registry of the work "Akarys91"? http://www.safecreative.org/work/1003015656515	
7	**Pre–Glossary of Terms**	
	a. Terms from Competency Questions (CQ)	
	Work (26.15%) *License (15.39%)* *Right (6.15%)* *Author (7.69%)* *Work types (21.54%)* *Special Condition (3.08%)*	*License Creator (3.08%)* *Jurisdiction (3.08%)* *Registry (4.61%)* *Rights holder (4.61%)* *License holder (3.08%)* *Informer (1.54%)*
	b. Terms from Answers	
	Rights (4%) *Licenses (20%)* *Work types (12%)* *Work (28%)* *Informer (4%)*	*Jurisdiction (4%)* *Author (16%)* *License Creator (8%)* *Rights holder (4%)*

Autores 3.0			
Registry Ontology			
7	**Pre–Glossary of Terms**		
	c. Objects		
		Rights	
		Display Right Reproduction Right Public Performance Right Transformation Right Communication Right Distribution Right	
		Special Conditions	
		Commercial use. Non-Commercial use. Attribution. Countered Downloads. Embedded. Limitation of exclusive rights. Miscellanea. Non-Military uses. Non-Nuclear uses. Derivative works must be licensed under the same license of the original one (Share-alike).	
		Work Types	
		Audio Choreography Database Drawing/Plan/Map Photography	Score Sculpture Software Text Video

Autores 3.0		
Registry Ontology		
7	**Pre–Glossary of Terms**	
	c. Objects	
		Licenses
		Apache BSD GFDL (*GNU Free Documentation License*) GPL (*General Public License*) Creative Commons Licenses: CC-BY, CC-BY-NC, CC-BY-NC-ND, C-BY-NC-SA, CC-BY-ND y CC-BY-SA. L-GPL (*Lesser General Public License*) MIT Mozilla Public License Personalized Licenses
		Jurisdictions
		Canadian, Spanish , etc.
		People
		Author, Rights holder …

Table 1 – Registry ontology requirements specification

Reused Resources

Three ontological resources have been reused during the development of this ontology. These resources have also been reused to develop the Basic ontology, however less classes and properties were reused to develop the latter ontology because the Registry ontology is an extended version of the Basic ontology. Classes, object properties and data properties of each reused ontological resource will be shown, showing specifically using a continuous red rectangle which classes and properties have been reused by both the Basic and Registry ontologies and with a discontinuous red rectangle which resources have been used exclusively by the Registry ontology.

First of all, it has been reused the FOAF [1] (acronym of *Friend of a friend*) ontology. This ontology was created to describe people, their activities and their relationships with people or things. Illustration 1 shows the classes defined in this ontology.

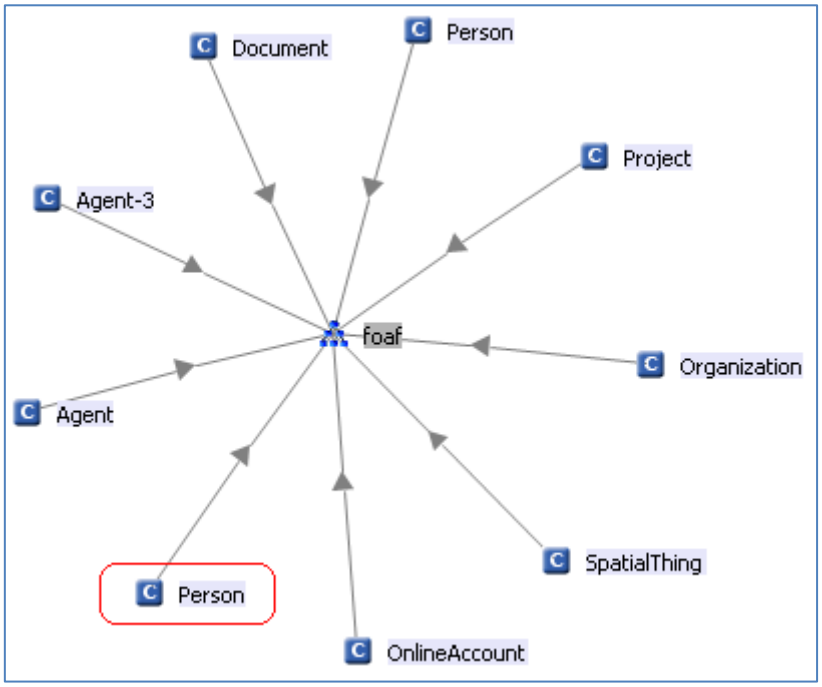

Illustration 1 – Classes defined in the FOAF ontology

The class Person has been reused by the Registry ontology (also by the Basic ontology) to be able to represent authors, informers, rights holders and licensees. Illustration 2 shows the object properties defined in the FOAF ontology.

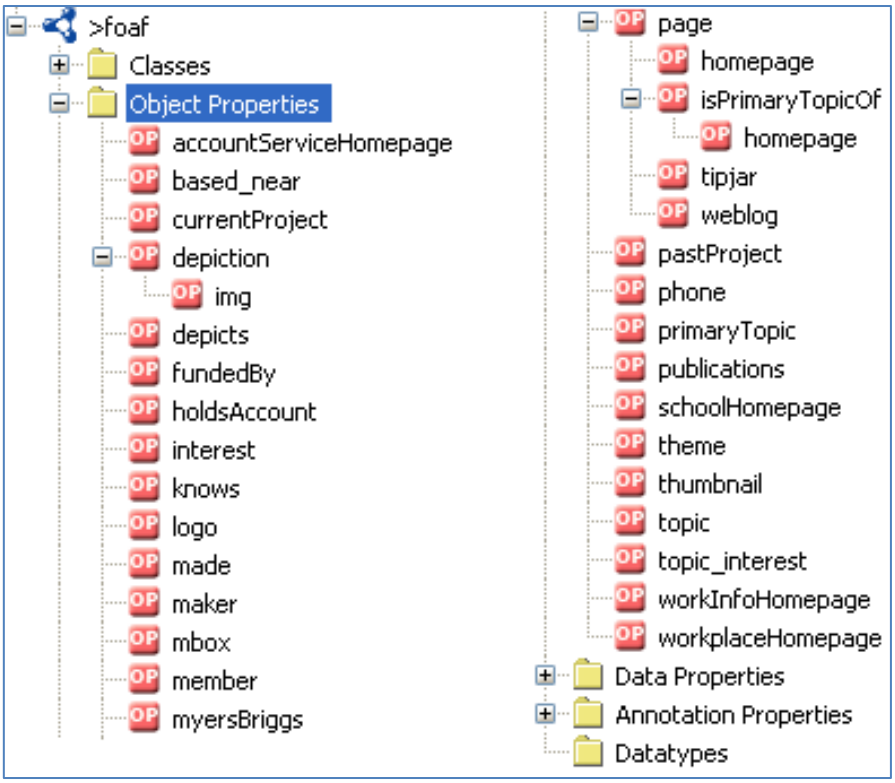

Illustration 2 – Object properties defined in the FOAF ontology

The Registry ontology has not reused any object property like the Basic ontology because being able to relate an author with other objects is beyond the goals of this project.

Lastly, Illustration 3 shows the data properties defined by the FOAF ontology. This type of properties is useful to describe people (to describe creators in our case).

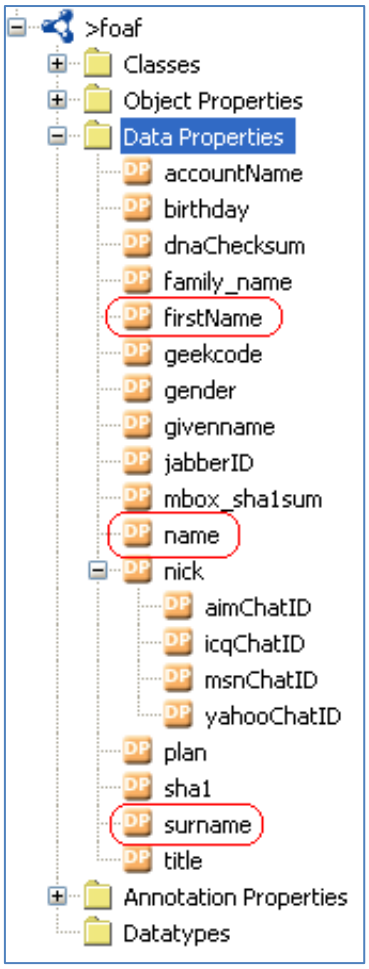

Illustration 3 – Data properties defined in the FOAF ontology

The properties *name*, *firstName* and *surname* have been reused to represent the full name, first name and surname of the creator. The remaining data properties have not been reused because its use would exceed the goals of this project.

Secondly, it has also been reused the Simple Dublin Core (DC) ontology [2]. This ontology was created in order to describe any type of information resource and it allows describing online resources to make them easily traceable. The object properties defined by this ontology are shown in Illustration 4.

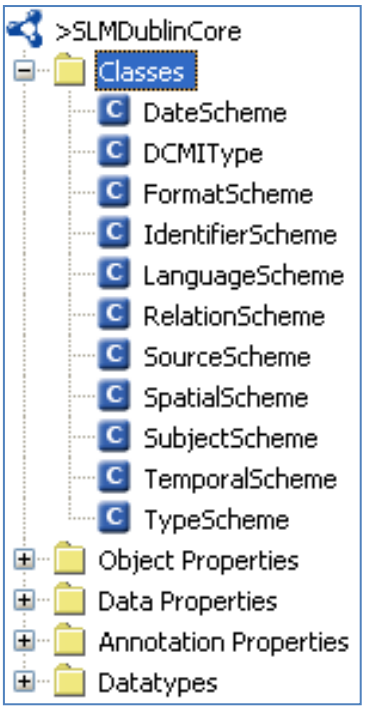

Ilustración 4 – Clases de la ontología Dublin Core básica

These classes are very generic and therefore they are not useful for the ontology to be built. The object properties defined by this ontology are shown in Illustration 5.

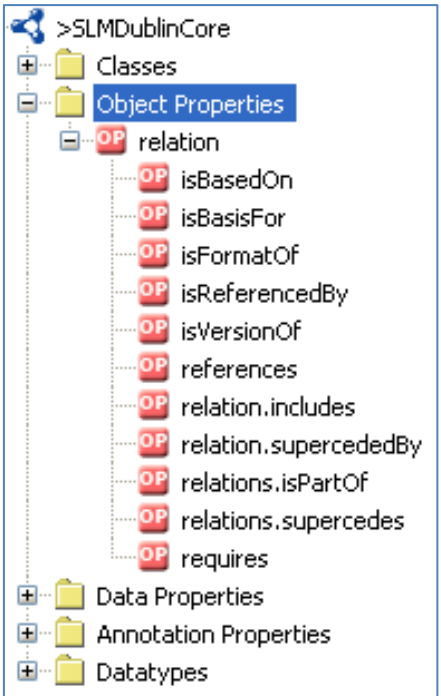

Illustration 5 – Object properties defined in the Simple DC ontology

As can be seen, none of these object properties have been reused due to the fact that it is beyond the goals of this project to model the relationships between works. The data properties defined by this ontology are shown in Illustration 6.

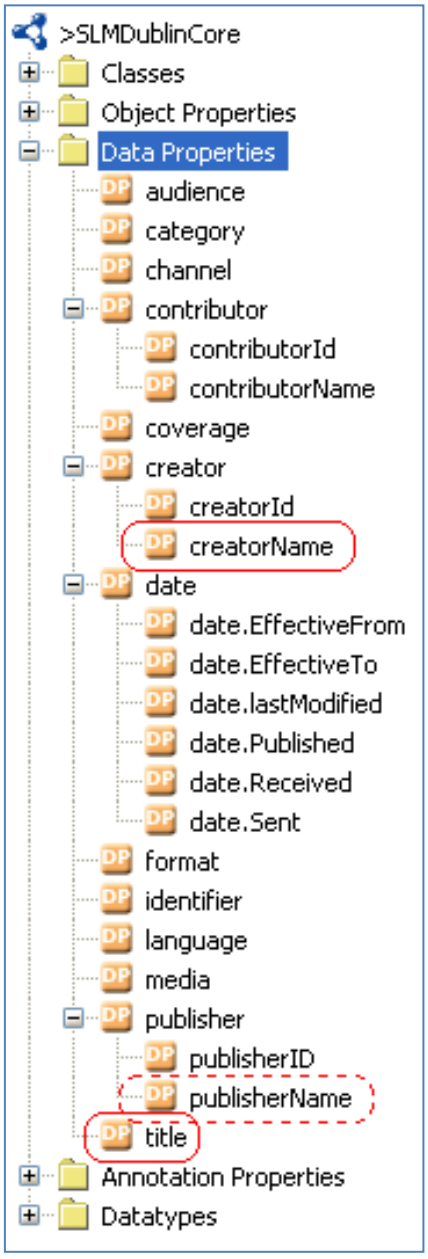

Illustration 6 – Data properties defined in the Simple DC ontology

The properties *creatorname* and *title* have been reused to represent the creator's full name and the title of the work. It could be said that the property *creatorname* of the Simple DC ontology is equivalent to the property *name* of the FOAF ontology.

The Registry ontology have also reused the property *publisherName*, so it can now represent the licensee's name, that is, the person who has published the work and, therefore, has registered it under a certain license.

Thirdly, the ontology *Qualified Dublin Core* [3] has also been reused. The *Qualified DC* ontology is an extension of the *Simple DC* ontology. The main difference between them lies in the definition of the classes *Audience*, *Provenance* and *RightsHolder*, that allow representing whom a work has been published for or could be useful to, where the work comes from and who holds their rights. The latter class is especially relevant for our project.

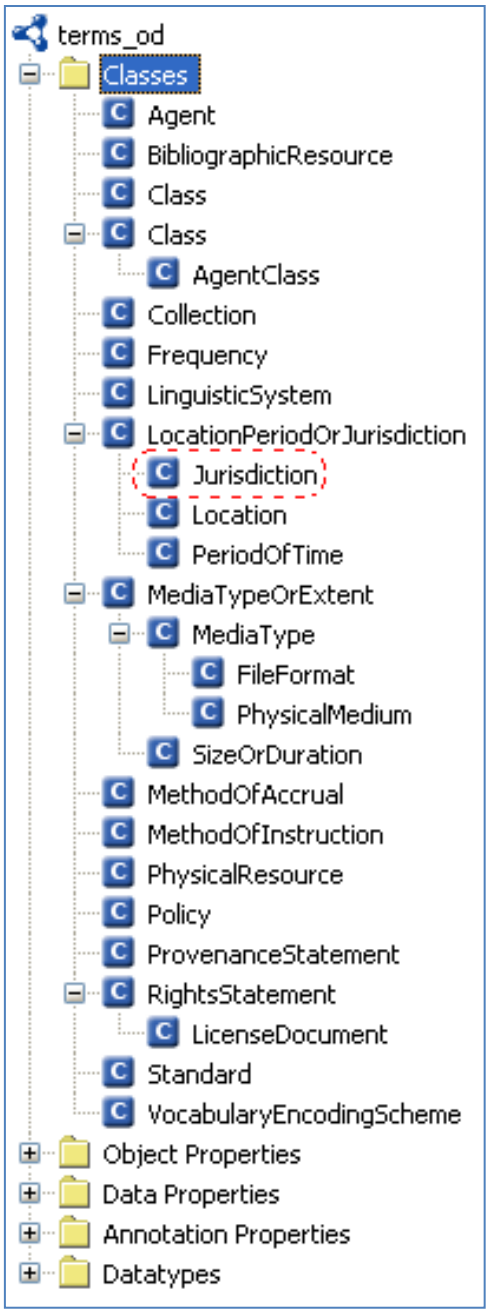

Illustration 7 – Classes defined in the Qualified DC ontology

The Registry ontology, contrary to the Basic ontology, uses the class *Jurisdiction* (that is important to model the intellectual property of a work). Therefore, it could be possible to represent the jurisdiction in which a work was created.

Illustration 8 shows the object properties of this ontology.

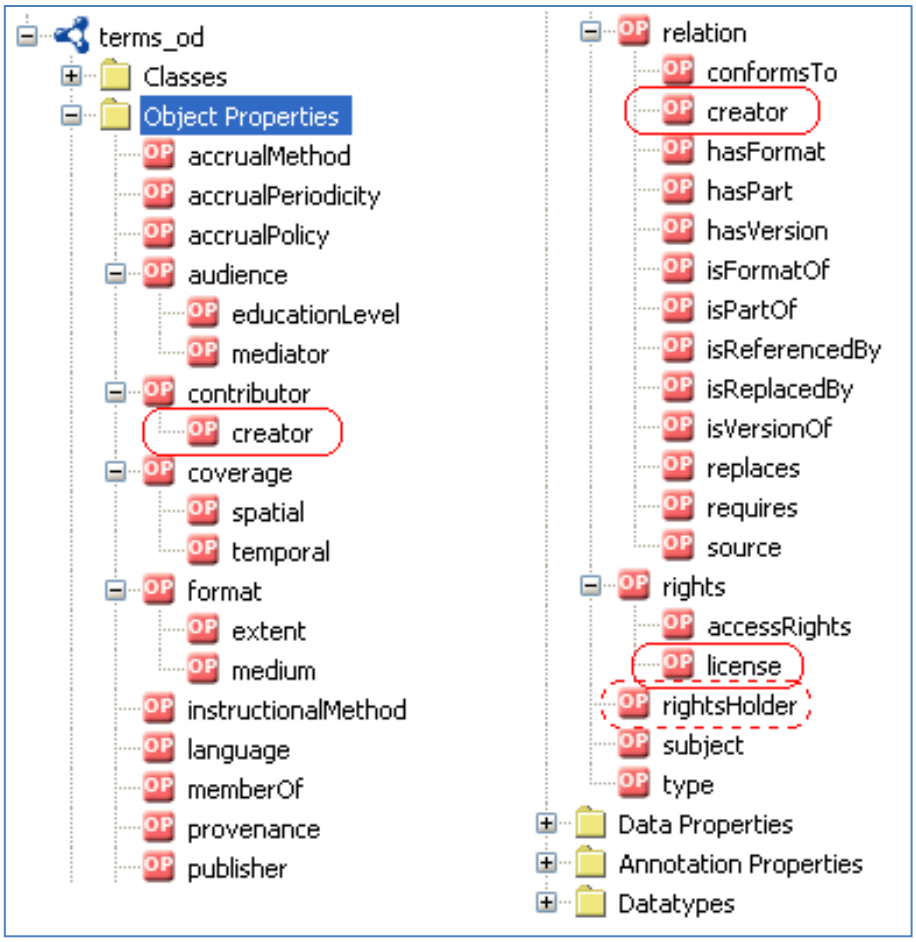

Illustration 8 – Object properties defined in the Qualified DC ontology

The properties *creator* are equivalent and are used to relate each work with its creator, while property *license* relate *Work* and *License* objects and will indicate which license a work is registered under.

The Registry ontology also reuses the property *rightsHolder* to show who the rights holder of each work is (it wasn't necessary to take into account in the Basic ontology). Therefore, while the Basic ontology shows the class *Person* identifying the author or creator (original rights holder of a work by law), the Registry ontology adds the property *rightsHolder* to distinguish the cases when the creator has transmitted (by means of inter vivos or mortis cause acts according to the general rules of law) the exploitation rights of a work. In this case, the *rightsHolder* authorizes the use of a work to third users by means of a license or contract and it appears in the registry.

Illustration 9 shows the data properties defined by this ontology.

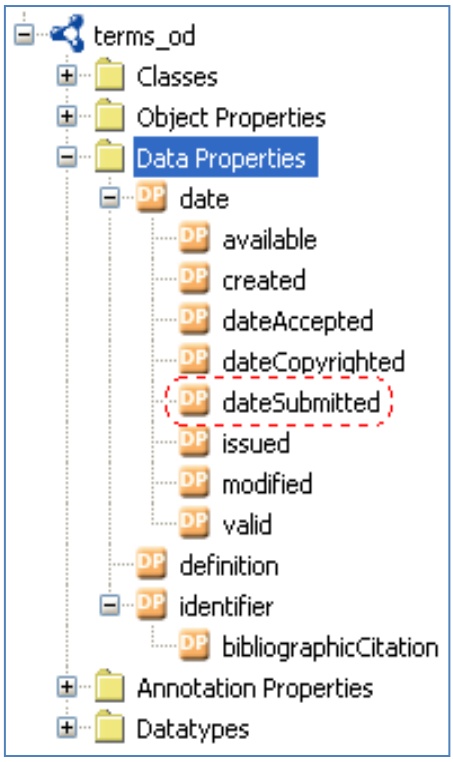

Illustration 9 – Data properties defined in the Qualified DC ontology

The property *dateSubmitted* is used by the Registry ontology, contrary to the Basic ontology, to indicate the date in what the work was registered.

Conceptual Model

Illustration 10 shows the importation hierarchy of the ontology network generated by *Neon Toolkit*. It can be seen that *registro* is the main ontology.

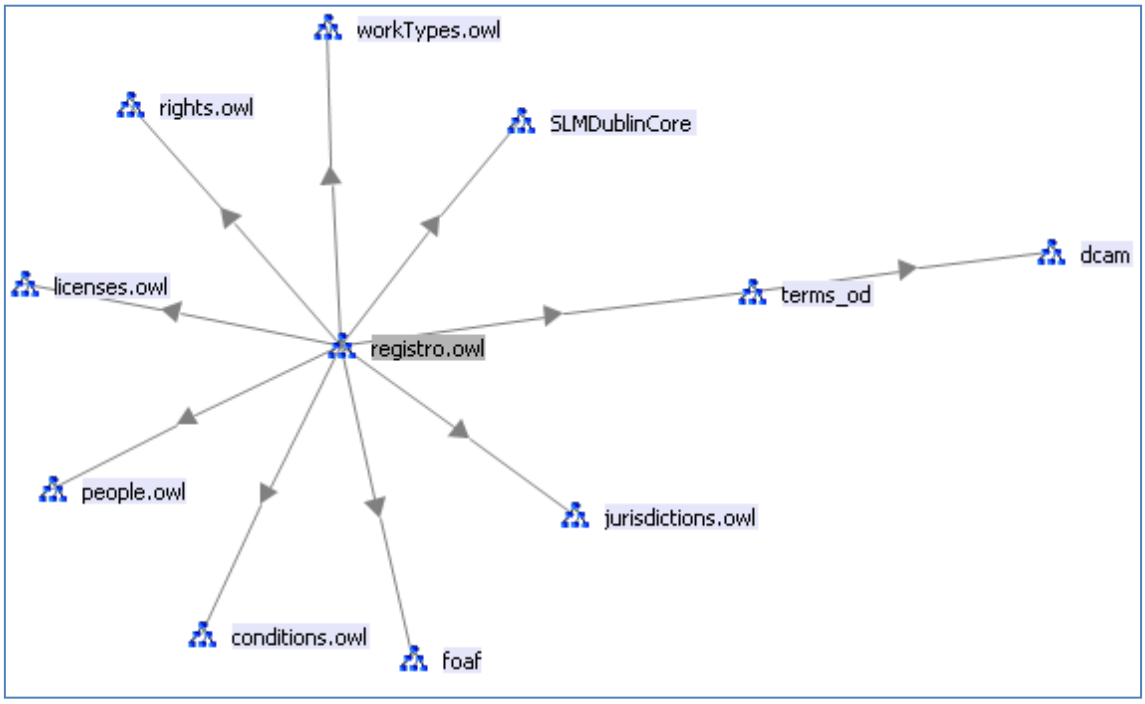

Illustration 10 – Importation hierarchy of the ontology network

The Registry ontology models authors, work types, licenses, rights and special conditions. However, this ontology is not populated with individuals and — to obtain a more modular result — the ontologies *licenses*, *workTypes*, *rights*, *people* and *conditions* are populated with the remaining individuals and are imported by the *registro* ontology. Besides, the *jurisdictions* ontology contains class *Jurisdiction* and it has been populated in a semiautomatic way with 192 individuals that match with the existing jurisdictions in the international context[2].

Registro is the main ontology and imports the remaining ones, including *jurisdictions*. It also imports the *FOAF* [1], *Simple Dublin Core* [2] and *Qualified Dublin Core* [3] ontologies — due to the fact that some of their resources have been reused.

The ontology has been developed in English, documented in English and Spanish and fulfills the 31 specified requirements.

[2] Jurisdictions list is very similar to nationalities list and has been obtained in [6].

Illustration 11 shows the class diagram[3] of the Registry ontology. A better understanding of the ontology can be achieved with a class diagram because it explicitly shows how its classes relate each other.

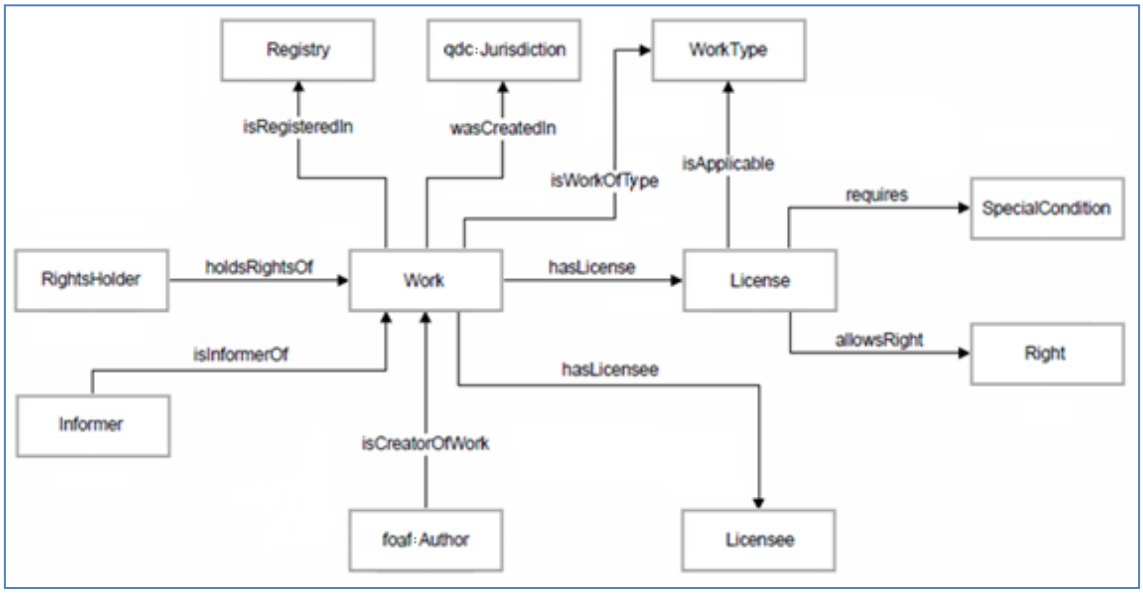

Illustration 11 – Class diagram of the Registry ontology

The main classes of the ontology are *Work* and *License*. A work is registered in a registry, has been created by an author in a specific jurisdiction, an informer has informed about (the intellectual property of) a work, someone handles its rights and somebody is the titular of a license that at the same time is applicable to some work types, allows some rights and requires some special conditions.

The object properties shown in the class diagram — together with its inverse properties — are the more relevant defined in this ontology. On the other hand, the data properties to define author's personal data, name and URL of the registry, title of the work and URL where the full legal text of the license can be found are the most relevant ones.

The most relevant individuals defined in this ontology are defined in ontology *Jurisdiction* with 192 jurisdictions, together with the individuals for each type of person, license, special condition and right taken into account in this ontology.

[3] Rectangles represent classes and directed arrows between classes indicate that they are related through and object property whose names are on the arrows. The origin of the arrow is the domain of the property and is directed towards the class that is the range of the property.

Classes *Author*, *RightsHolder*, *Informer* and *Licensee* appearing in the diagram of Illustration 11 are part of the taxonomy of class *Person*.

Illustrations 12 to 16 show the hierarchies created for classes *License*, *Person*, *Right*, *Special Condition* and *WorkType*, respectively.

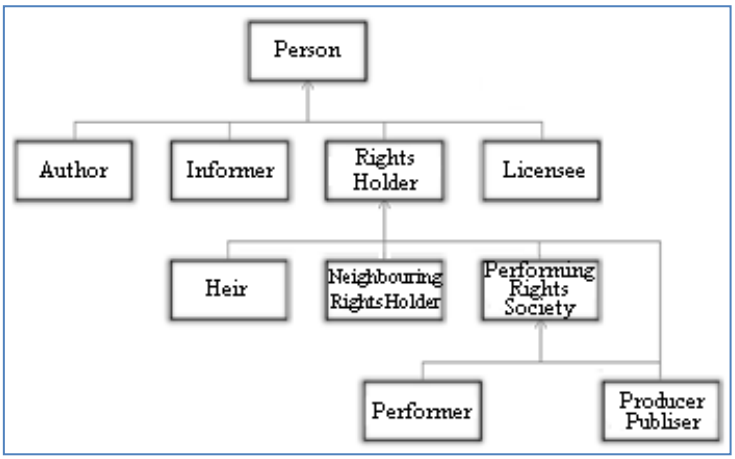

Illustration 12 – Taxonomy of the class Person

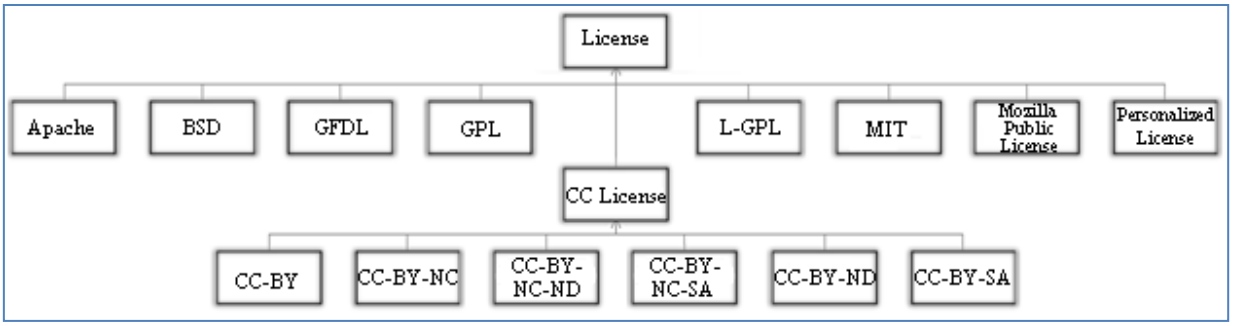

Illustration 13 – Taxonomy of the class License

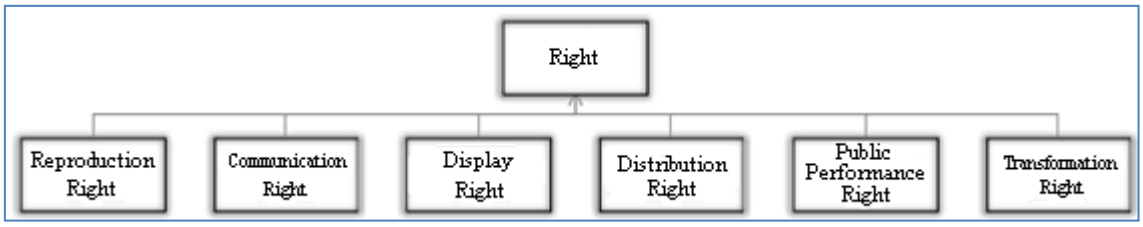

Illustration 14 – Taxonomy of the class Right

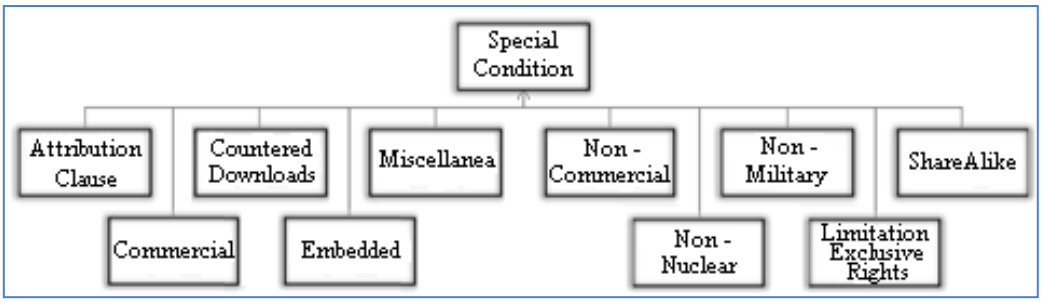

Illustration 15 – Taxonomy of the class Special Condition

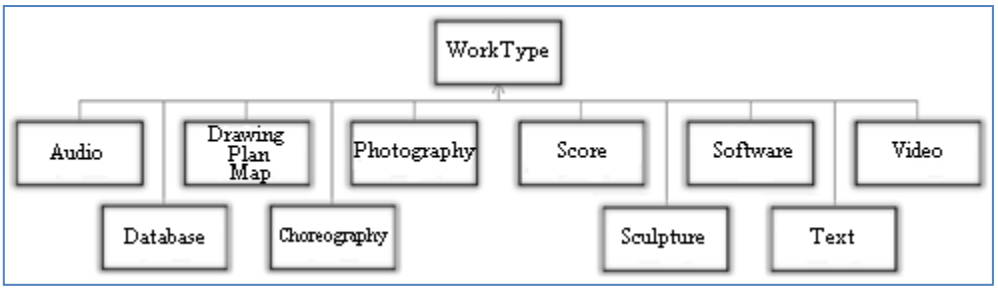

Illustration 16 – Taxonomy of the class WorkType

Implementation

The ontology has been developed using *NeOn Toolkit* [4] v3.2.1 and *gOntt*, *Relationship Visualization*, *RaDON* and *XD Tools* plug-ins and some documentation[4] has also automatically generated using *OWLDoc* plug-in.

Several screenshots of the Registry ontology from the *NeOn Toolkit* will be shown in this section. Besides, OWL code will also be shown to provide a more detailed view about this ontology.

The five ontologies signaled in Illustration 17 have been reused. *Core*, *SLMDublinCore* and *foaf* ontologies have been reused directly, while *dcam* and *terms_od* have been reused indirectly due to the fact that these ontologies are imported by the former ones.

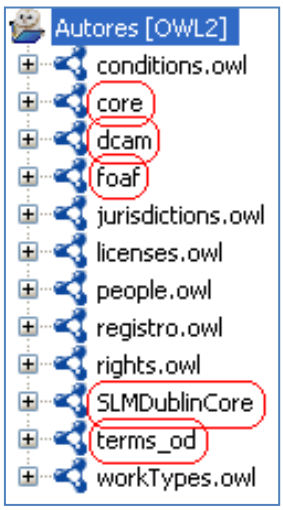

Illustration 17 – Reused ontologies in the Registry ontology

Illustration 18 shows the classes defined by the Registry ontology (those that have been reused are signaled).

[4] The documentación is available in [6].

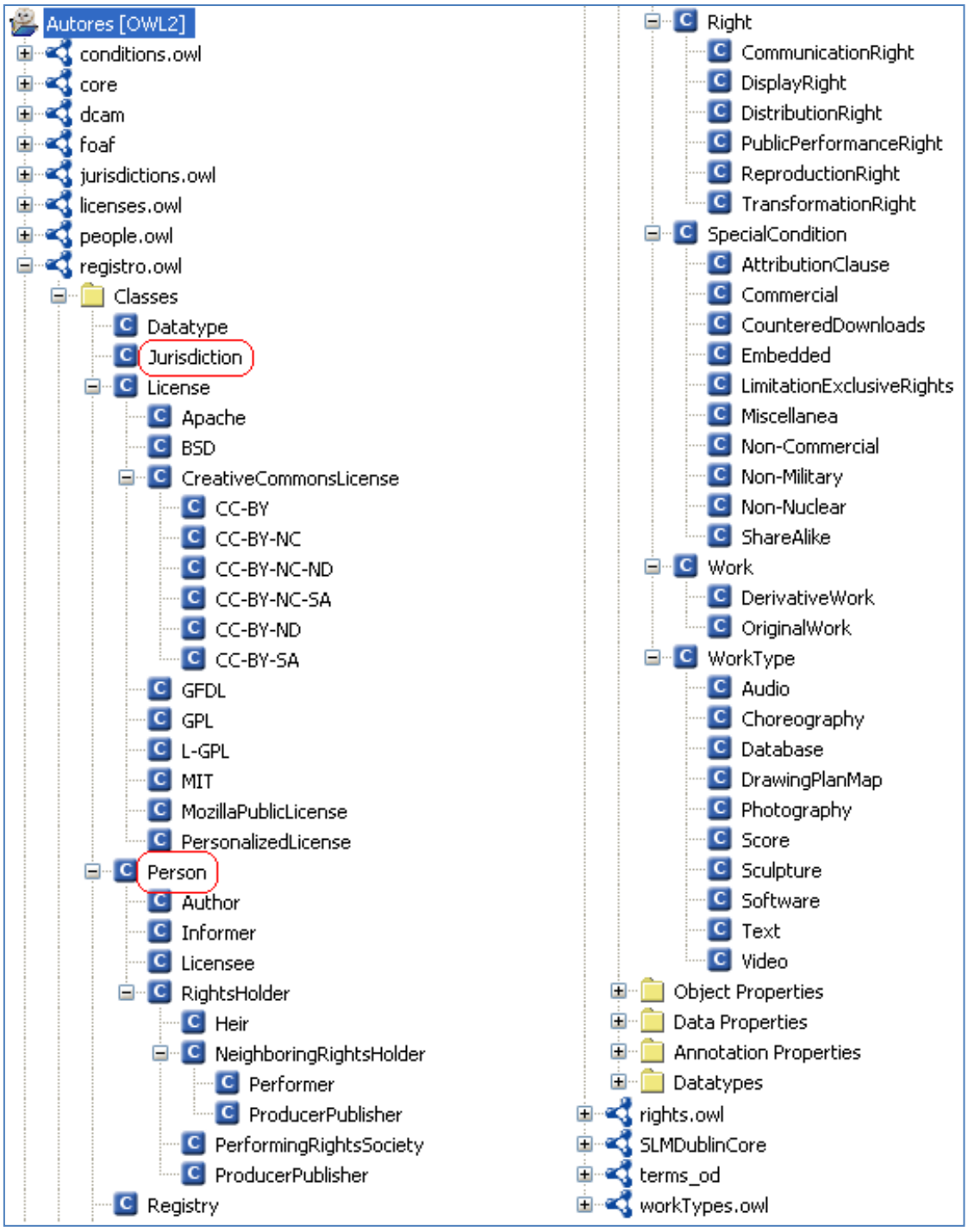

Illustration 18 – Classes defined in the Registry ontology

Illustration 19 shows the data properties defined by the Registry ontology (those that have been reused are signaled).

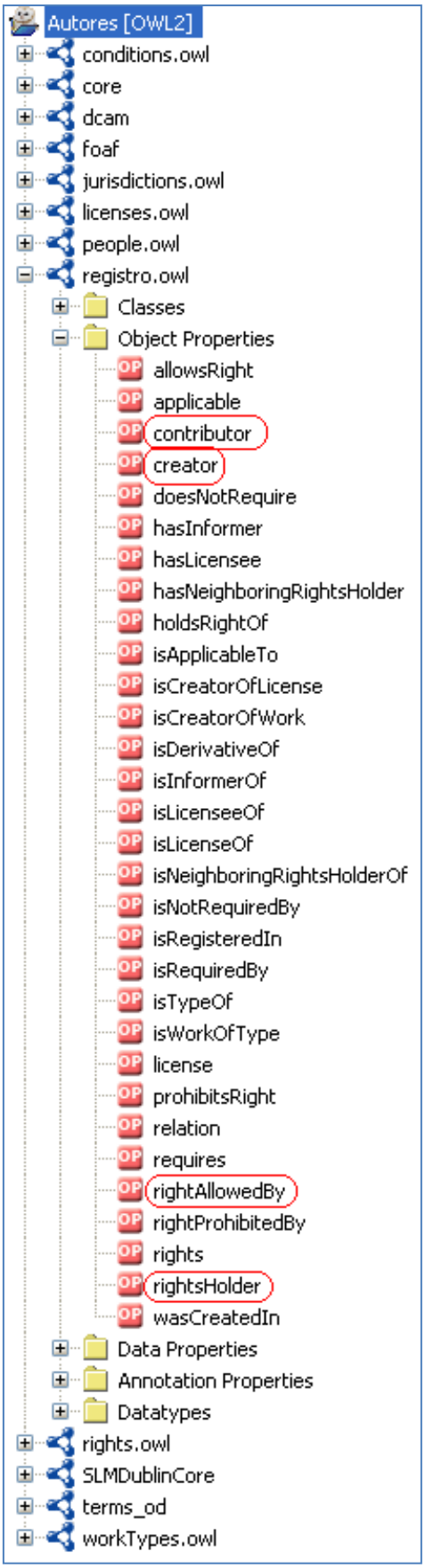

Illustration 19 – Data properties defined in the Registry ontology

Illustration 20 shows the object properties defined by the Registry ontology (those that have been reused are signaled).

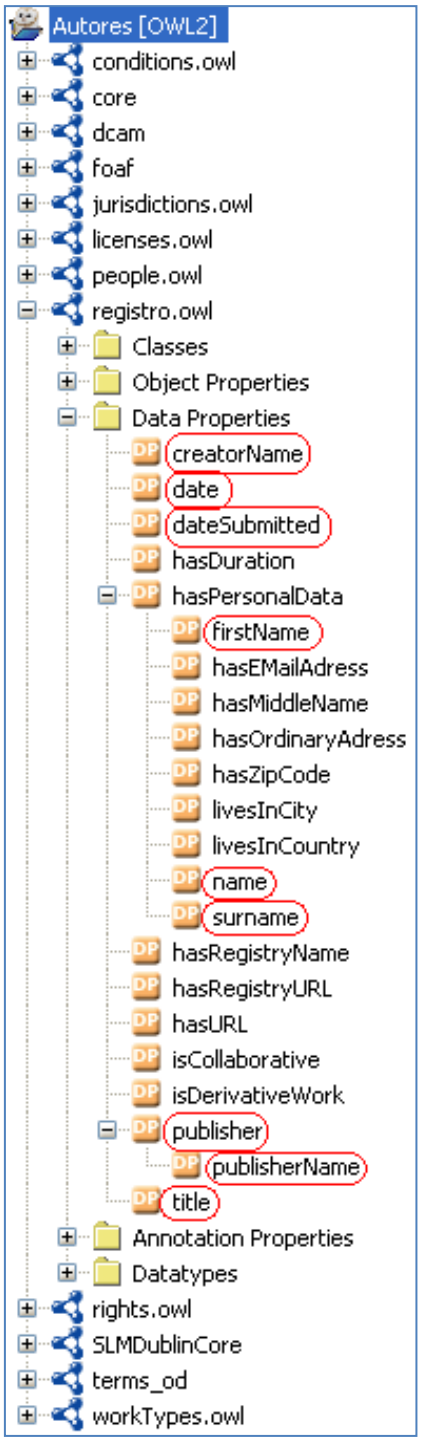

Ilustración 20 – Propiedades de datos de la ontología Registro

Firstly, Illustration 21 shows the relationships of the class *Work*.

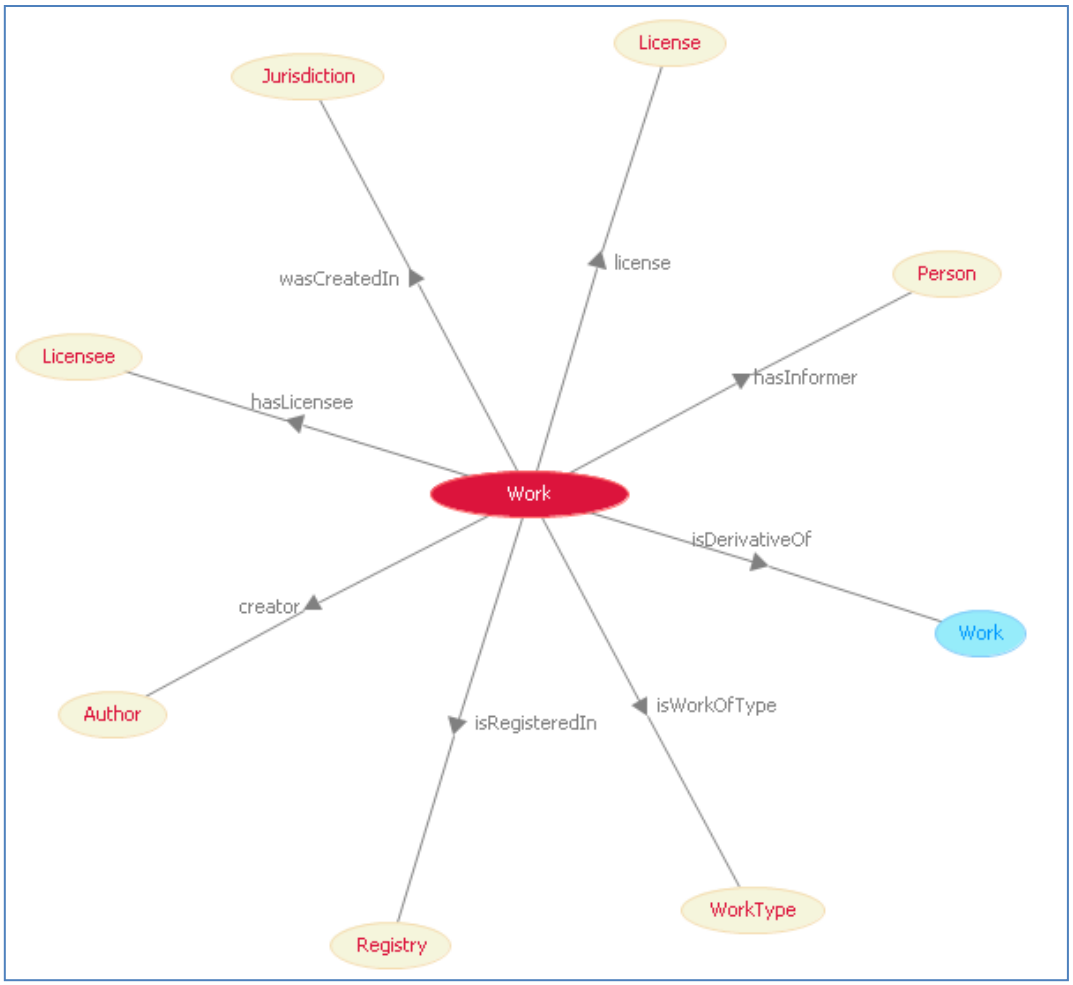

Illustration 21 – Relations of the class Work in the Registry ontology

It can be observed that a work is registered under a certain license, is registered in a specific registry and was created by a specific author. But, contrary to the Basic ontology, it is also taken into account the jurisdiction in which a work was created, its type, who licensed it, who is the rights holder and who informed about its rights.

Next relationships of the class *License* are shown in Illustration 22.

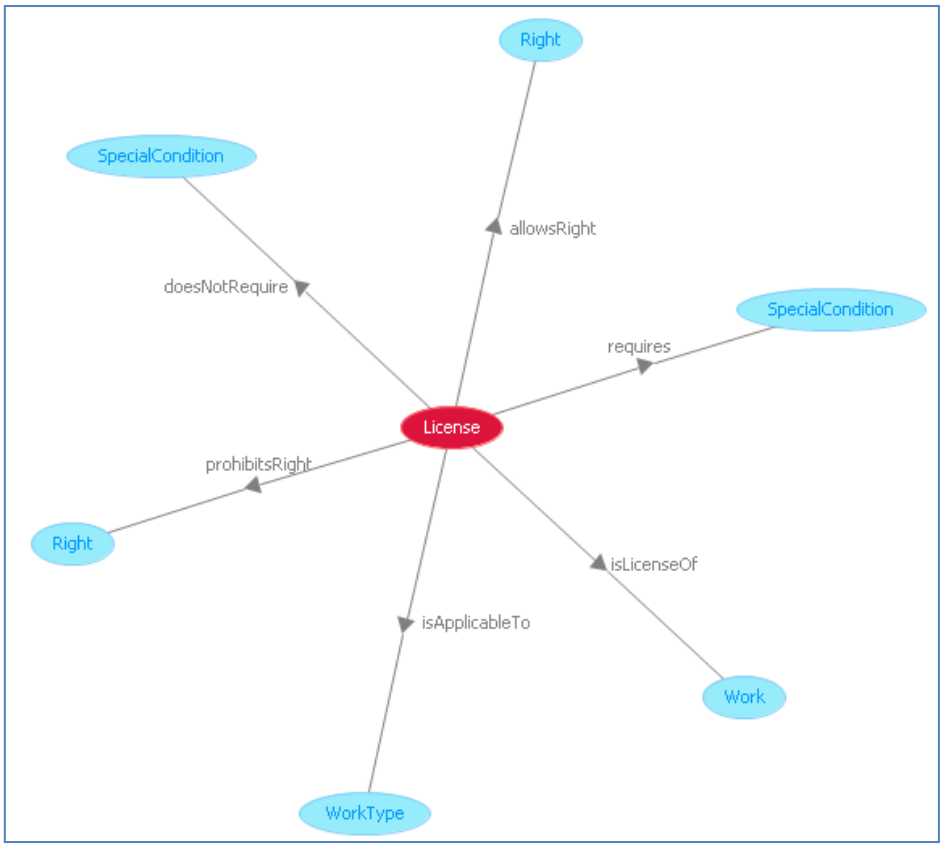

Illustration 22 – Relations of the class License in the Registry ontology

It can be observed that a work is licensed under a specific license and each license allows certain rights and requires some special conditions. What is more, and contrary to the Basic ontology, it is specified to which types of work a license can be applied. Lastly, due to the close world assumption, it is explicitly indicated, which special conditions are not required and which rights are not allowed.

Illustration 23 shows the Manchester Syntax of the class *License*.

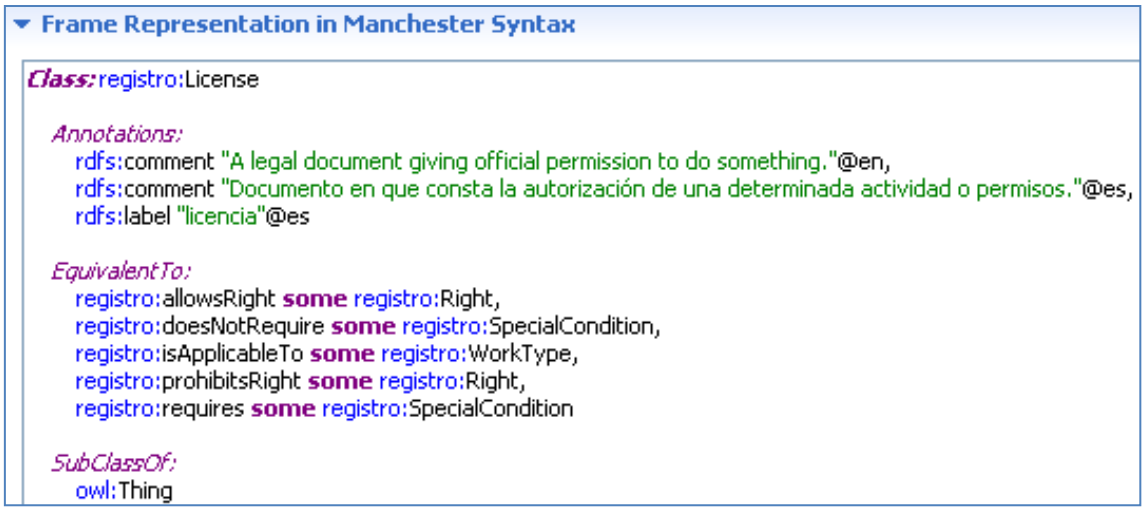

Illustration 23 – Class License in the Registry ontology using the Manchester syntax

It can also be observed that not only this class is documented with a definition both in English and Spanish and has class expressions to model that a license allows some rights and requires certain special conditions, but also it defines a fifth class expression to indicate that it is only applicable to certain work types.

Lastly, it is shown in Illustration 24 the OWL code of the class *License* where the fifth class expression can be observed (as well as those who model which rights does a license allow and which special conditions does a license require).

```
<owl:Class rdf:about="#License">
    <rdfs:label xml:lang="es">licenciatario</rdfs:label>
    <owl:equivalentClass>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#isApplicableTo"/>
            <owl:someValuesFrom rdf:resource="#WorkType"/>
        </owl:Restriction>
    </owl:equivalentClass>
    <owl:equivalentClass>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasSpecialCondition"/>
            <owl:someValuesFrom rdf:resource="#SpecialCondition"/>
        </owl:Restriction>
    </owl:equivalentClass>
    <owl:equivalentClass>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#allowsRight"/>
            <owl:someValuesFrom rdf:resource="#Right"/>
        </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
    <rdfs:comment xml:lang="en"
        >A legal document giving official permission to do something.</rdfs:comment>
    <rdfs:comment xml:lang="es"
        >Documento en que consta la autorizaci&#243;n de una determinada actividad o permisos.</rdfs:comment>
</owl:Class>
```

Illustration 24 – Simplified OWL code of the class License in the Registry ontology

Illustration 25 shows the individuals of the *Jurisdiction* ontology, consisting of the 192 existing jurisdictions in the international context.

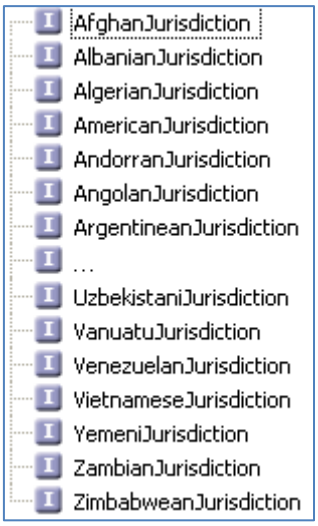

Illustration 25 – Individuals of the Jurisdictions ontology

Registro ontology contains no individuals and the *licenses*, *workTypes*, *rights*, *people* and *conditions* ontologies contain the individuals of the subclasses of the classes *License*, *WorkType*, *Right*, *Person* and *SpecialCondition*, respectively. Besides, these individuals have been linked, using the label *sameAs*, with a set of individuals from *DBpedia* [8] and *Freebase* [9], agreed by the community and equivalent to some of them.

Illustration 26 shows the individuals of the ontology *licenses*, that contains the individuals of the subclasses of the class *License*, that is, *Apache*, *BSD*, etc. The individuals linked with *sameAs* have been signaled.

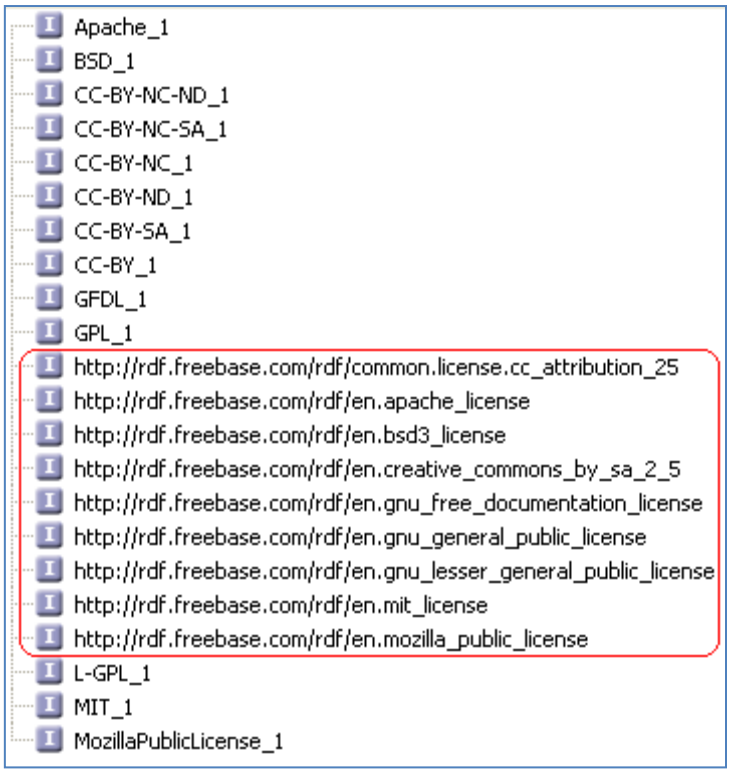

Illustration 26 – Individuals of the ontology *licenses*

Illustration 27 shows the individuals of the ontology *rights*, that contains the individuals of the subclasses of the class *Right*.

Illustration 27 – Individuals of the ontology *rights*

Illustration 28 shows the individuals of the ontology c*onditions*, that contains the individuals of the subclasses of the class *SpecialCondition*.

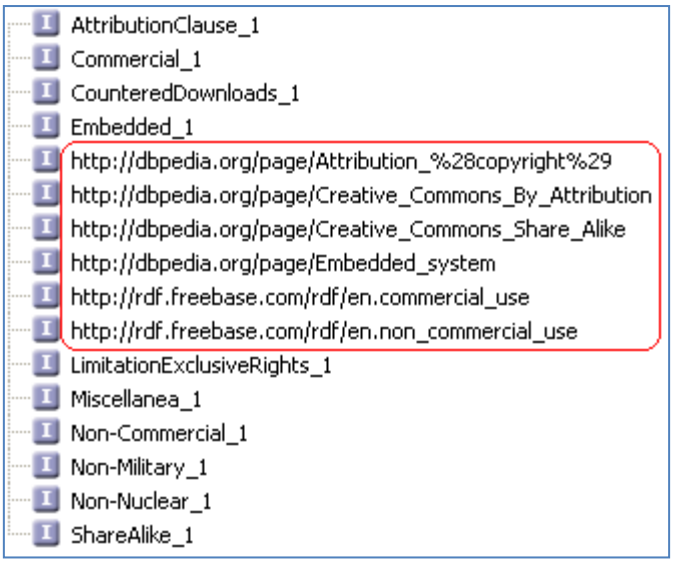

Illustration 28 – Individuals of the ontology *conditions*

Illustration 29 shows the individuals of the ontology *workTypes*, that contains the individuals of the subclasses of the class *WorkType*.

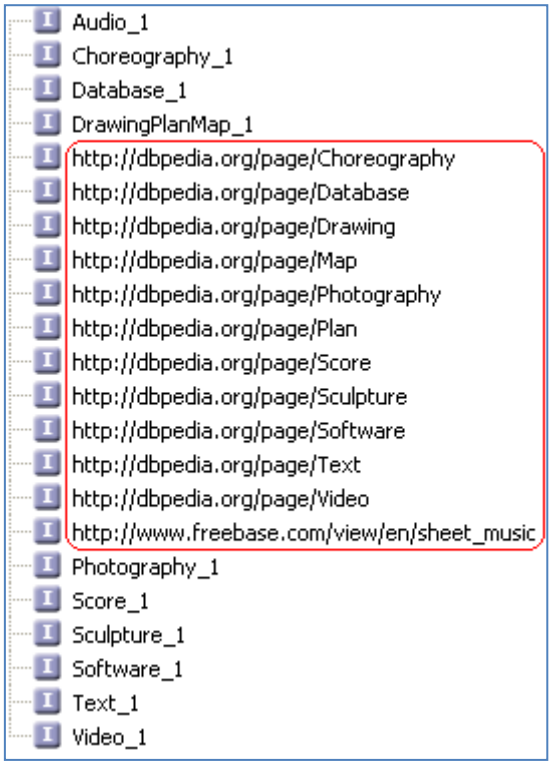

Illustration 29 – Individuals of the ontology *workTypes*

Illustration 29 shows the individuals of the ontology *people*, that contains the individuals of the subclasses of the class *Person*.

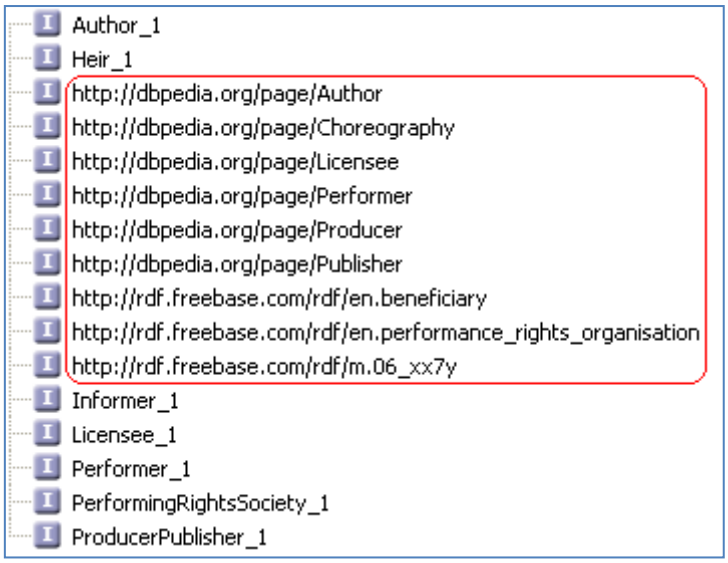

Illustration 30 – Individuals of the ontology *people*

Illustration 31 shows the relationships of the individual *Audio_1*.

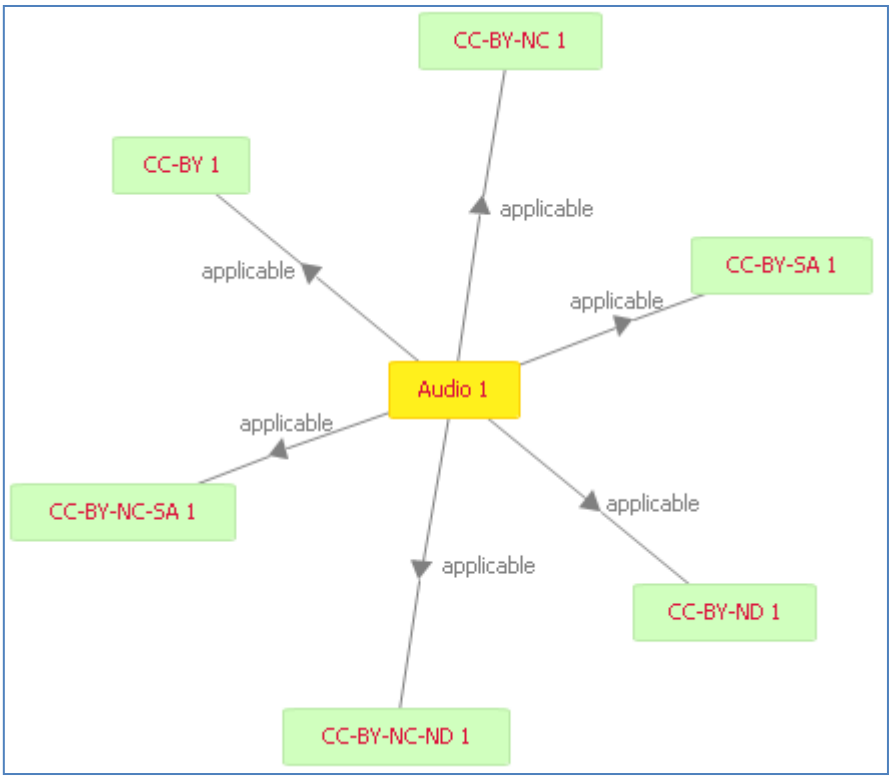

Illustration 31 – Relationships of the individual Audio_1

Illustration 32 shows the relationships of the individual *Comercial_1*.

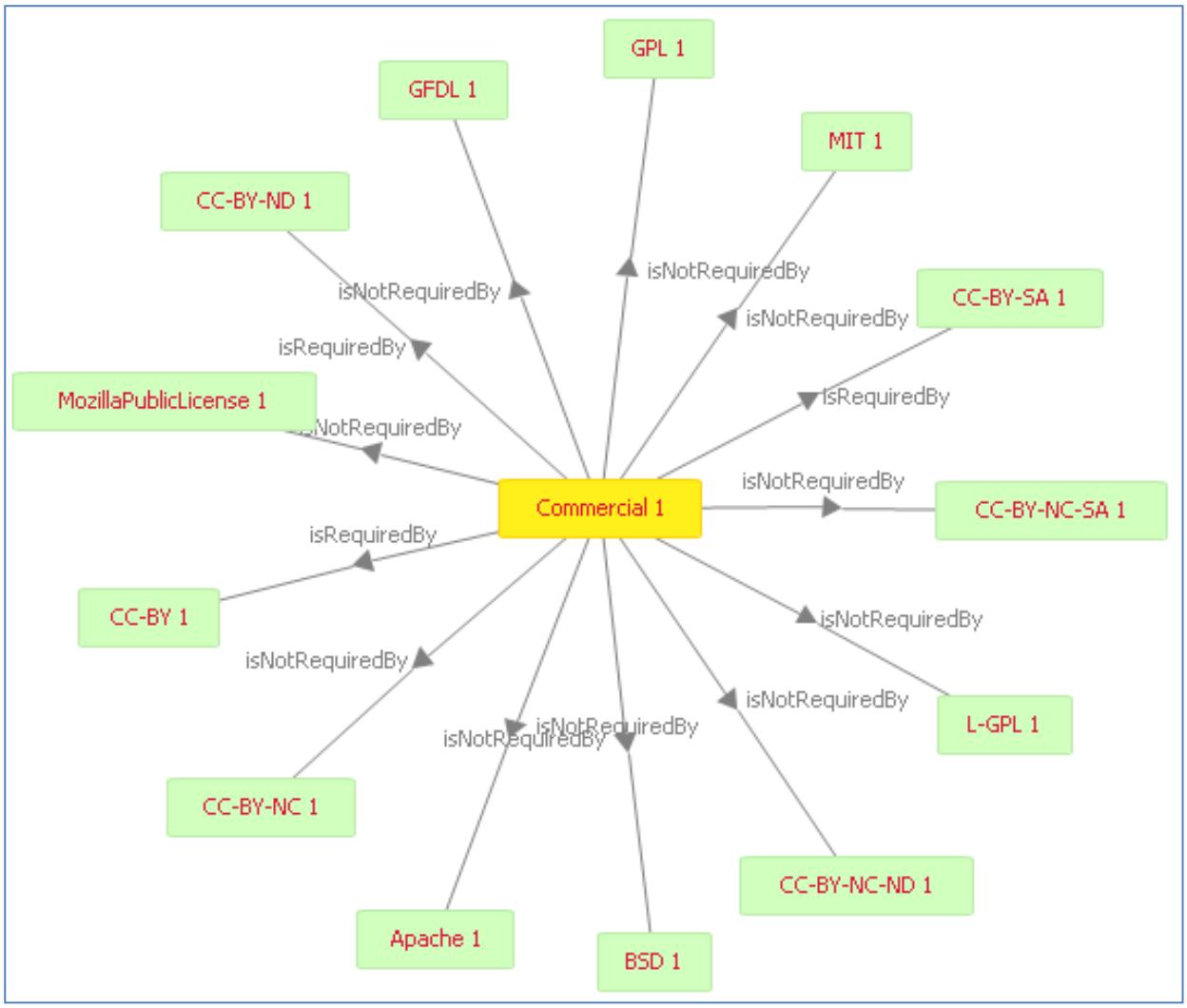

Illustration 32 – Relationships of the individual Commercial_1

Illustration 33 shows the relationships of the individual *DistributionRight_1*.

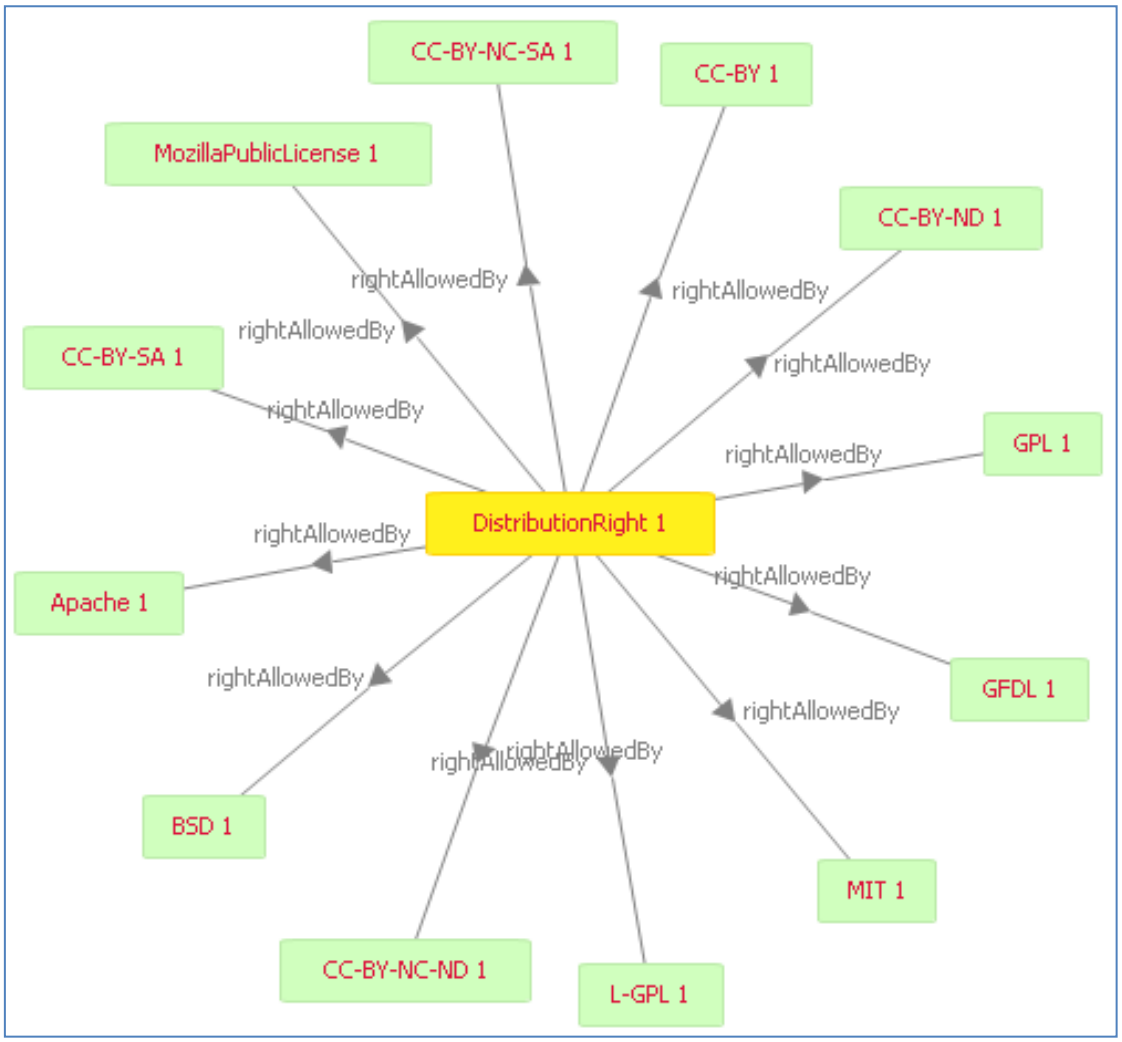

Illustration 33 – Relationships of the individual DistributionRight_1

Illustration 34 shows the relationships of the individual *CC-BY-NC-ND_1*.

Illustration 34 – Relationships of the individual CC-BY-NC-ND_1

It can be observed that this license (the most restrictive *Creative Commons* license) allows of the rights (taken into account in the ontology) but the transformation right (*No Derivative*) and assuming no commercial uses (*Non-Commercial*). Besides, with the Registry ontology it can be known which work types this license can be applied to (this license can be applied to every single work type considered in this project).

Illustration 35 shows the Manchester Syntax of the individual *CC-BY-NC-ND_1* and shows that it has been inherited from classes *CC-BY-NC-ND* and *Thing*.

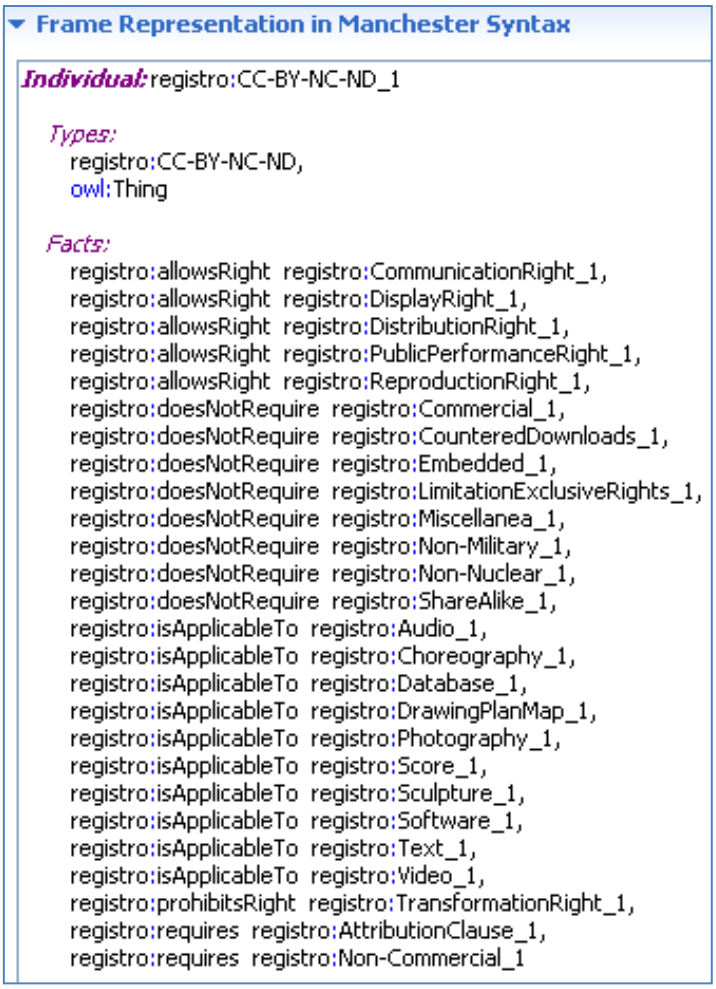

Illustration 35 – Individual CC-BY-NC-ND_1 in the Registry ontology using the Manchester Syntax

It can be seen again that this syntax shows the work types this license is applicable to. The OWL code of the individual CC-BY-NC-ND_1 is shown in Illustration 36 (properties *doesNotRequire* and *prohibitsRight* have been removed due to legibility reasons).

```
<owl:Thing rdf:about="#CC-BY-NC-ND_1">
    <rdf:type rdf:resource="#CC-BY-NC-ND"/>
    <isApplicableTo rdf:resource="#Audio_1"/>
    <isApplicableTo rdf:resource="#Choreography_1"/>
    <allowsRight rdf:resource="#CommunicationRight_1"/>
    <isApplicableTo rdf:resource="#Database_1"/>
    <allowsRight rdf:resource="#DisplayRight_1"/>
    <allowsRight rdf:resource="#DistributionRight_1"/>
    <isApplicableTo rdf:resource="#DrawingPlanMap_1"/>
    <hasSpecialCondition rdf:resource="#Non-Commercial_1"/>
    <isApplicableTo rdf:resource="#Photography_1"/>
    <allowsRight rdf:resource="#PublicPerformanceRight_1"/>
    <allowsRight rdf:resource="#ReproductionRight_1"/>
    <isApplicableTo rdf:resource="#Score_1"/>
    <isApplicableTo rdf:resource="#Sculpture_1"/>
    <isApplicableTo rdf:resource="#Software_1"/>
    <isApplicableTo rdf:resource="#Text_1"/>
    <allowsRight rdf:resource="#TransformationRight_1"/>
    <isApplicableTo rdf:resource="#Video_1"/>
</owl:Thing>
```

Illustration 36 – Simplified OWL code of the individual CC-BY-NC-ND_1 in the Registry ontology

Illustration 37 shows the Registry ontology statistics.

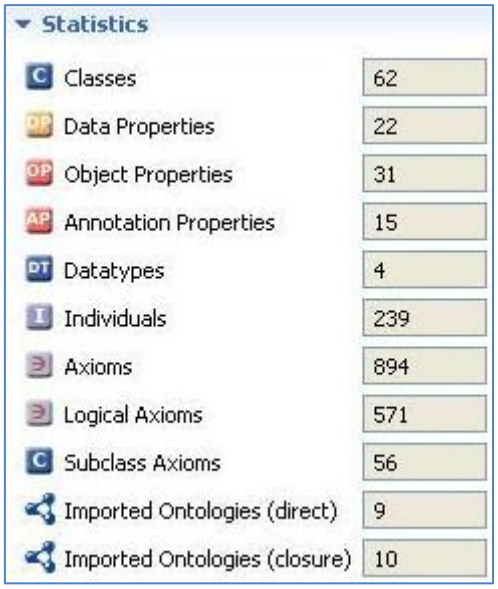

Ilustración 37 – Estadísticas de la ontología Registro

As can be shown, the ontology has 62 classes, 22 data properties, 31 object properties and 239 individuals.

Lastly, it is worth mentioning that this ontology has been evaluated by the *RaDON* plug-in, stating that it is a coherent a consistent ontology.

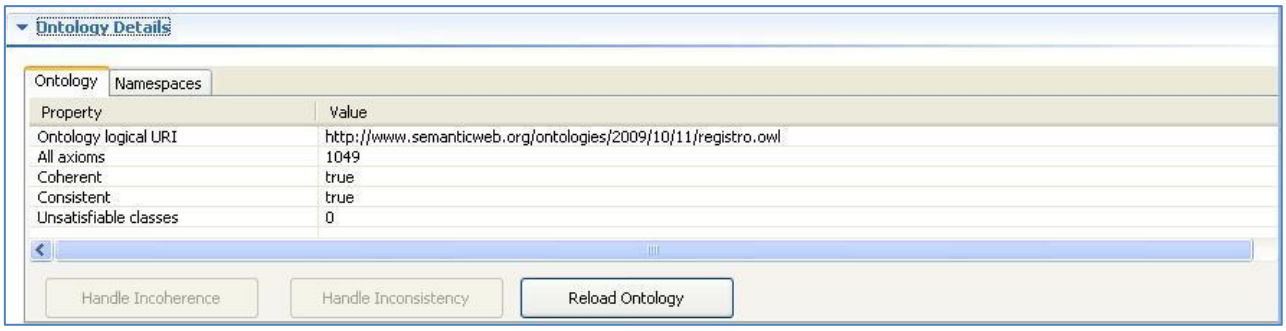

Illustration 38 – Registry Ontology evaluation using *RaDON* plug-in

References

[1] http://www.mindswap.org/2003/owl/foaf

[2] http://ontologies.semanticarts.com/dublincore/dublincore.owl

[3] http://purl.org/NET/dc_owl2dl/terms_od

[4] Neon Toolkit Wiki. http://neon-toolkit.org/wiki/Main_Page

[5] Suárez–Figueroa, M. C., Gómez–Pérez, A., and Villazón–Terrazas, B. 2009. How to Write and Use the Ontology Requirements Specification Document. Lecture Notes In Computer Science, vol. 5871. Springer–Verlag, Berlin, Heidelberg, 966–982.

[6] Nationalities list. http://gist.github.com/raw/274449/999d690210ea300d9fd2e4886db8c68ec8887965/nationalities.txt

[7] Semantic Copyright. http://semanticcopyright.org/

[8] DBpedia. http://dbpedia.org/About

[9] Freebase. http://www.freebase.com/

[10] WordNet. http://wordnet.princeton.edu/