Improving ODRL 2.2: current limitations and theoretical solutions

Andrea Cimmino[1,*,†], Nicoletta Fornara[2,†]

[1] Universidad Politécnica de Madrid, Madrid
[2] Università della Svizzera italiana, Switzerland

Abstract

The Open Digital Rights Language (ODRL) is a W3C recommendation widely adopted to express permissions and prohibitions on digital content and services in multiple domains, including the Internet of Things, European Data Spaces, and Knowledge Graphs. Despite its wide adoption, the ODRL recommendation includes only an ontology and nothing regarding the enforcement of the policies. Due to this reason, multiple limitations, gaps, and issues have been reported over time. In this article, common issues and limitations of ODRL reported in the literature and others identified by the authors are analysed and grouped in a proposed taxonomy for categorising them. Based on these findings, the authors propose a set of improvements and theoretical solutions. These proposals outline a foundation for addressing the main challenges and inform future enhancements to ODRL.

Keywords

ODRL policies, ODRL semantics, ODRL limitations

1. Introduction

The Open Digital Rights Language (ODRL) is a W3C recommendation to define policies for the use of content and services [1]. It has been widely adopted, or recommended, in different domains like Internet of Things [2, 3], European Data Spaces [4, 5], Solid PODs [6], or Knowledge Graphs [7]. ODRL policies allow to express permissions, prohibitions, or duties in different context such as usage control or data access. However, from the practical point of view, due to the use cases requirements ODRL's practitioners have lean towards expressing policies for access control instead of usage control. As a result, most of the policies expressed and the scenarios described revolve around permissions or prohibitions.

Implicitly, a policy or norm has been endowed with a shared meaning, so that it should be widely understood, and because of this meaning it should be possible to be evaluated in order to know if it was fulfilled or violated. This implicit nature of the policies implies two key dimensions: on the one hand, the *descriptive semantics* of the policies must be unequivocally and universally interpreted by a person or software system (agent, code library, etc.); on the other hand, policies are not mere readable documents, but they have a *behavioural dimension* that can be evaluated thanks to the formal semantics of the language in which they are expressed. In the literature, these dimensions have been addressed differently: A) The adoption of ontologies, and particularly standard ones, allows one to have a wide shared meaning to unequivocally and universally define and understand concepts; however, ontologies are only descriptive and cannot be evaluated *per se* [1, 8]. B) Expressing policies as rules within rule systems [9, 10, 11, 12] allows to evaluate policies; however, these policies do not have the shared meaning that ontologies provide. As a result, the usage of policies in practical scenarios requires understanding and taking into account these two dimensions.

ODRL and Beyond: Practical Applications and Challenges for Policy-base Access and Usage Control. OPAL 2025 Co-located with the Extended Semantic Web Conference, Portorož, Slovenia · June 1 2025.

*Corresponding author.
†These authors contributed equally.
andreajesus.cimmino@upm.es (A. Cimmino); nicoletta.fornara@usi.ch (N. Fornara)
https://portalcientifico.upm.es/es/ipublic/researcher/307657 (A. Cimmino); http://usi.to/dte (N. Fornara)
0000-0002-1823-4484 (A. Cimmino); 0000-0003-1692-880X (N. Fornara)

From the first version of ODRL to the latest, the standard has promoted an ontology (which has evolved over time) to express its policies, but has not published a recommendation related to the behavioural nature of the policies. In all versions of ODRL, its semantics is described informally in English and no formal specification is provided, except a draft on the semantics of ODRL [13] showing the relevance of the topic for the ODRL Community Group. The adoption of this standard in practical scenarios and its comparison with other policy models have reported limitations and drawbacks [14] due to, namely, ontology limitations and the lack of documents related to the evaluation of ODRL policies. As a result, proposals have focused on ontology issues [15, 16, 17, 18, 19, 11, 20, 21], or on ODRL enforcement/evaluation from the theoretical point of view as formal semantics [22, 23, 24, 15], or from the practical point of view with implementations that might be supported with formal semantics [25, 6, 11, 20, 10, 17, 26, 27]. In addition, to the authors' knowledge, only one proposal has explored the usage control scenario exploring obligations through ODRL policies [11].

In this article, the authors analysed and classified common problems reported in the literature regarding ODRL and presented a set of proposals to improve the current version, i.e. ODRL 2.2. On the one hand, the limitations and issues reported around ODRL, and new ones identified by the authors, are analysed, taxonomized, and specific proposals for addressing them are provided. On the other hand, a deeper analysis of the obligations expressed using ODRL is performed based on use cases not covered by the current ODRL model, and a set of improvements for ODRL are proposed. The findings of this article may set the foundations for solving several of the current ODRL issues that have been reported and that hinder the practical adoption of ODRL. However, note that the scope of this article is to propose practical solutions, but not provide an implementation for each. Their implementation will be addressed in future contributions. This paper is organised as follows. In Section 2 a taxonomy of the limitations of ODRL is presented, then in Section 3 some theoretical approaches or ideas are presented for solving the problems discussed. Finally, in Section 4 some conclusions are drawn.

2. Limitations of ODRL

As mentioned in the Introduction, the policies require two types of semantics, on the one hand, the *descriptive semantics* that must convey a shared meaning and that must be unequivocally and universally interpreted. On the other hand, *behavioural semantics* that must allow to evaluate a policy with respect to a state of the world, obtaining the same results regardless of the system used to evaluate the policy. Figure 1 shows the taxonomy of limitations proposed in this article, and in the following subsections the authors will delve into the different categories, identifying the particular limitations from the related works, and proposing new ones.

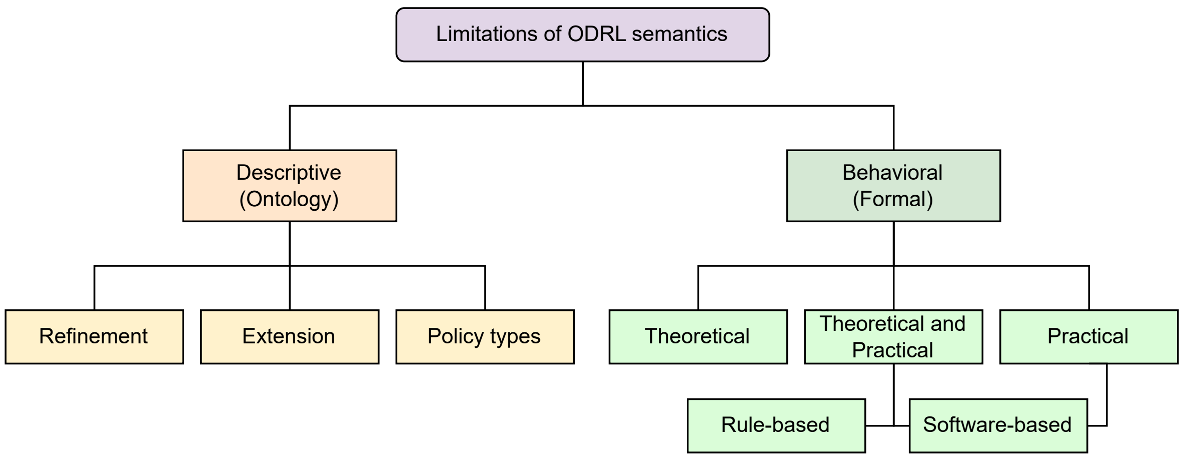

Figure 1: Taxonomization of ODRL limitations and issues

2.1. Descriptive Semantics (Ontology)

The limitations and issues related to the descriptive semantics of ODRL revolve around the definition or conceptualisation of its ontology. The authors differentiate different categories:

Refinement limitations: articles in this category report issues derived from the lack of precision, or accuracy, in certain terms in the ODRL ontology. Some of these terms are operands (left and right), operators, or actions that are defined in a too broad sense, leading to multiple interpretations of them when implemented in practical scenarios [15, 16, 17]; namely, due to the fact that the term itself has a deeper, or more complex, definition in natural language that is not present in the ontology. For example, the following RDF excerpt from the ODRL ontology shows how the operand *odrl:dateTime* is defined:

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

:dateTime
    a :LeftOperand, owl:NamedIndividual, skos:Concept ;
    rdfs:isDefinedBy odrl: ;
    rdfs:label "Datetime"@en ;
    skos:definition "The date (and optional time and timezone) of exercising the action of the Rule. Right
         operand value MUST be an xsd:date or xsd:dateTime as defined by [[xmlschema11-2]]."@en ;
    skos:note "The use of Timezone information is strongly recommended. The Rule may be exercised before (
        with operator lt/lteq) or after (with operator gt/gteq) the date(time) defined by the Right
        operand. Example: dateTime gteq 2017-12-31T06:00Z means the Rule can only be exercised after (and
         including) 6:00AM on the 31st Decemeber 2017 UTC time."@en ;
    skos:scopeNote "Non-Normative"@en .
```

Listing 1: Definition of the *odrl:dateTime* term extracted from the ODRL ontology[1]

Note that several restrictions are stated in natural language as the value of the data property *skos:definition* and *skos:note*. These restrictions encode information that does not only affect the term itself, but also others such as *odrl:lt*, *odrl:lteq*, *odrl:eq*, *odrl:gt*, *ordl:gteq*, or *odrl:neq* and the values of the *right operand* when used in conjunction with *odrl:dateTime*. All this information must be expressed in the ontology so that the term *odrl:dateTime* and its meaning can be unequivocally understood and used by humans, and also machines; since this is precisely the goal of using an ontology. Otherwise, this term could be implemented in multiple different ways. This issue applies to all the terms related to actions, operands, and operators in the ODRL ontology.

Extension limitations: articles in this category report issues found when applying ODRL in some use cases where the terms of the ODRL ontology are too broad or generic for that specific domain [18, 19]. In particular, these issues are not derived from a lack of precision in the definition of the term but rather due to the need to have domain-specific terms. A common approach to this issue followed by other ontology standards like SAREF [28], is to define a core ontology that is general and broad and then develop extensions linked to the core ontology for specific domains. For example, SAREF is about devices that make measurements and then depending on the domain 12 modules are defined, one of them is SAREF4BLDG [29] where concepts such as buildings, geospatial features and specific devices such as sensors or actuators are defined.

In this sense, ODRL provides a recommendation to extend the domain specificity of policies by creating the so-called profiles [6, 30]. However, this mechanism is insufficient to ground the ODRL ontology to a particular domain. Also, sometimes the profiles are used to refine operators and operands, which is also insufficient. OWL already provides extension mechanisms [31] and well-known methodologies, such

[1] https://www.w3.org/ns/odrl/2/ODRL22.ttl

as NeOn [32], to adapt core ontologies [32], such as ODRL, to specific domains by defining extension modules that extend the core ontology with terms belonging to this specific domain. The usage of these mechanisms, rather than defining new ones, is the approach followed by several standard ontologies from different domains such as SAREF [28], DCAT [33], or Brick [34].

Policy types: articles in this category report that ODRL is not expressive enough to describe certain types of policy and, therefore, require adding new concepts and relationships to the core ontology of ODRL [11, 20]. In [21] a list of different types of policies has been proposed. Crucial types of policies are those that are activated: by the occurrence of an event (e.g., someone pays for a certain service), by the satisfaction of a state of affairs (e.g., the actor of the regulated action is in a certain country), or both. Other types of policy are those that contain time constraints for the actions they regulate or for the period in which they are in force, such as prohibitions in force for an interval of time. Regarding the activation condition, in ODRL it is not possible to specify that an obligation or a prohibition is activated as a result of performing an action or by the occurrence of an event. For example, a policy may state that when users play a file, they are obliged to write a review.

Regarding temporal constraints, it is impossible to specify using the ODRL ontology the deadline by which an obliged action must be performed. For example, the obligation to delete a photo within 2 weeks of receiving it. In ODRL, it is not enough to specify a deadline using a *odrl:dateTime* refinement for the obliged action. This is because it is not clear how the *odrl:dateTime* property has to be evaluated. Differently, the deadline of an obligation should be used to represent a particular time event, before which the obligation can be fulfilled, and after which, if the obligation has not been fulfilled, it becomes violated; therefore, the notion of deadline should be represented in the ODRL ontology.

Another relevant problem of the ODRL ontology, in the context of policy types, is the usage of the *odrl:Duty* class to represent two different concepts. The first is the notion of an obligation for an agent to execute an action. This is represented in ODRL by connecting an individual that belongs to the *odrl:Duty* class to a policy by using the *odrl:obligation* property. The second is the notion of duty used as a precondition that must be satisfied in order to obtain a valid permission to perform an action. This is represented in ODRL by connecting an individual that belongs to the *odrl:Duty* class to a permission by using the *odrl:duty* property.

2.2. Behavioural semantics (formal)

The issues of behavioural semantics are related to the broad sense of evaluating a policy with respect to the state of the world. For this topic, ODRL lacks a standard document that describes its behavioural semantics. In the literature, it is possible to differentiate different categories:

Theoretical: articles in this category report the lack of formal semantics for ODRL and provide it [22, 23, 24, 15]. Note that these proposals do not provide any implementation to support the formal semantics stated in the articles.

Practical: articles in this category report the lack of an implementation in ODRL to evaluate policies and provide a software-based implementation without describing formal semantics or providing a conceptualisation or theoretical framework [25, 6].

Theoretical & Practical: articles in this category aim at providing a theoretical framework to evaluate policies and an implementation to support it. To this end, two main approaches exist: rule-based systems [11, 20, 10]; or software-based systems [17, 26]; which can (i) evaluate a certain subset of policies expressed with ODRL [27], (ii) evaluate a subset with extension capabilities and therefore potentially all the ODRL policies, or (iii) evaluate ODRL-based policies that are not exactly express only with ODRL.

2.3. Relationships between categories

Note that the categories identified in the previous section can affect each other. For example, the fact that the ODRL operands are broadly defined in the ontology but detailed in natural language leads to proposals that provide formal semantics and implementations that behave differently for the same operand, i.e. *odrl:dateTime*. This is caused by the need of refining the ontology that makes one term in the ontology potentially translated into several different understandings of its implementation.

When applying ODRL in a scenario like GDPR the core terms are not enough, or if there is a need to express policy with terms related to time. In these cases a domain-specific extension is needed, as has been reported in the past[16].

Similarly, from the behavioural point of view, practitioners have different levels of odrl:permission. Let us rely on an example to show these differences: *in a hospital a patient creates a policy that only a specific doctor has permission to access his/her clinical history*. Some practitioners understand the permission as if the patient is dying, and no one except the doctor can access the clinical history to save the patient's life. Instead, other practitioners understand that under extreme circumstances another physician or a nurse can access the clinical history, keeping a record of this action, and later the policy will be evaluated to check whether the violation of the permission was legit or not. This is caused due to the lack of more specific policy types because in this case, both scenarios describe a permission, but in the ontology they could not be described under the same terms because a machine, from the behavioural point of view, needs to act differently depending on which scenario are.

3. Proposals for ODRL 3.0

Considering the different categories for which various problems have been reported, this section aims at providing theoretical solutions or ideas for solving the general issues so that they could be considered in the proposal of a new version of ODRL. To this end, we propose the following list of improvements that will be detailed in the following subsections:

- **Proposal 1.** Refine the operands, operators and actions in the ODRL ontology;
- **Proposal 2.** Introduce and describe the concept of state of the world as a Knowledge Graph;
- **Proposal 3.** Introduce in ODRL the concept policy status;
- **Proposal 4.** Extending ODRL to support the specification of various types of policies;
- **Proposal 5.** SPARQL-based evaluation rules;
- **Proposal 6.** Introduce the notion of policy templates & variables.

3.1. Proposal 1: ODRL ontology refinement

ODRL is not the first standard that deals with ontology terms that somehow encode certain behaviour such as operands, operators, and actions; in the past, the OGC GeoSPARQL initiative had to deal with the same issue [35]. In this standard, their ontology describes geometries in terms of points, lines, or polygons and a set of functions that can be computed with them, for example, to compute the distance between two points or to know if a point is inside a polygon. All these terms, the data concepts and the function ones, were defined in their ontology along with a set of rules to translate the ontological terms related to functions into SPARQL query functions that any engine could implement. Following this approach, any ontological term related to function could be used unequivocally and with the same meaning in any SPARQL engine.

Following the OGC GeoSPARQL initiative, our first proposal is that ODRL adopts the FnO function ontology in its core ontology [36], as it has been done for functions[2], to describe operators and operands (when they act as functions, such as *odrl:dateTime*) and to refine actions to potentially include functions instead of being a mere concept. To showcase the benefits of adopting this ontology, the following excerpt shows how the *odrl:dateTime* definition in Listing 2 could be improved by narrowing down its degrees of interpretation:

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
```

[2]https://opengeospatial.github.io/ogc-geosparql/geosparql11/functions.ttl

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix fno: <https://w3id.org/function/ontology#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

odrl:dateTime a odrl:Operand, owl:NamedIndividual, skos:Concept, fno:Function ;
  rdfs:label "Datetime"@en ;
  fno:expects ( odrl:datetime_utc_offset ) ;
  fno:returns odrl:datetime_output .

odrl:datetime_utc_offset a fno:Parameter ;
  fno:type xsd:integer ;
  fno:required false ;
  rdfs:comment "This parameter is the UTC offset applicable to the dateTime of the system" ;
  skos:prefLabel "offset"@en .

odrl:datetime_output a fno:Parameter ;
  fno:required true ;
  fno:type xsd:dateTime .

odrl:before a odrl:Predicate, fno:Function ;
  rdfs:comment "This predicate returns true when the first parameter has an xsd:dateTime that occurs
       before the second xsd:dateTime parameter; if odrl:strict_before_parameter is set to true the
       function returns false in the case both parameter are the same" ;
  fno:expects (
   odrl:before_parameter_1
   odrl:before_parameter_2
   odrl:strict_before_parameter  ) ;
  fno:returns odrl:preticate_output .

odrl:before_parameter_1 a fno:Parameter ;
  fno:type xsd:dateTime ;
  fno:required true ;
  rdfs:comment "This parameter is an xsd:dateTime value" ;
  skos:prefLabel "Starting datetime"@en .

odrl:before_parameter_2 a fno:Parameter ;
  fno:type xsd:dateTime ;
  fno:required true ;
  rdfs:comment "This parameter is an xsd:dateTime value" ;
  skos:prefLabel "Ending datetime"@en .

odrl:strict_before_parameter a fno:Parameter ;
  fno:type xsd:boolean ;
  fno:required false ;
  rdfs:comment "This parameter states if the comparison of dates is strict; in the sense of if strict is
       true the before function returns false whether two xsd:dateTimes values are the same, and returns
       true if they are the same, and stric is set false. By default, this parameter is considered true" ;
  skos:prefLabel "stric"@en .

odrl:preticate_output a fno:Parameter ;
  fno:required true ;
  fno:type xsd:boolean .
```

Listing 2: Definition of the dateTime term in the ODRL ontology with the FnO ontology

Listing 2 shows only an example of how ODRL could adopt this proposal to express operands, actions, and operators using the FnO ontology. However, it does not necessarily mean the best ontology design. Note that with the previous excerpt, the ontological definition of the operand can not be interpreted in different ways from the behavioural point of view and, also, the restrictions regarding the operator and right operand are encoded when used with the *odrl:dateTime* operand. In addition to refine the ontology following this approach, and mimicking the OGC GeoSPARQl initiative, an ontology linking these functions to the SPARQL language must be defined (in all probability, the one defined by GeoSPARQL could be reused[3]).

Note that operands can also be static values; however, as discussed by Cimmino et al. [17] these values could: i) lead to privacy leak if shown explicitly in the policies; or ii) these values may be dynamic and change over time. To address this issue, the concept of policy variable is introduced in Section 3.6. Although the adoption of FnO will bring benefits in terms of policy specification, it could also increase the complexity when writing the policy due to the need of writing more RDF triples in order to be as specific as possible when declaring the policy. In addition, the policies could potentially lean towards programming similar semantics defining several functions and their parameters.

3.2. Proposal 2: A Knowledge Graph as State of the World

Traditionally, in the policy literature, the state of the world represents the concepts that encode all the information a system needs to know to evaluate a policy [11]. In ODRL there is no definition of how this state of the world shall be expressed or represented. Due to this reason and the fact that ODRL policies are expressed in RDF, **it is crucial that the state of the world should be expressed as a Knowledge Graph in RDF following an ontology.**

This approach may entail several benefits from different point of view: a) policy consistency, since policies are already a Knowledge Graph in RDF, they could be linked directly to the Knowledge Graph of the state of the world; b) evaluation, processing the data needed by the policy in the same format of the policy eases the evaluation process; c) expressiveness, the state of the world could be expressed with any domain ontology providing rich representation of it; d) modularity, named graphs linked to one or more policies could be used to modularize the state of the world that relates to such policies; and e) storage, the Knowledge Graphs of the policies and the state of the world could be stored in the same place (centralised) or in different places (decentralised) without entailing any issue thanks to the nature of the Knowledge Graphs.

The following Listing 3 shows an example of policy and state of the world.

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix doc: <http://www.organization.com/documents/> .
@prefix per: <http://www.organization.com/people/> .
@prefix odrlx: <http://www.w3.org/ns/odrl/2-extension/> .

## Policy
<http://example.com/policy:33CC>
  a odrl:Agreement ;
  odrl:profile <http://example.com/odrl:profile:09> ;
  odrl:prohibition [
    odrl:action odrl:index ;
    odrl:assignee per:99 ;
```

[3]https://www.w3.org/ns/sparql-service-description.ttl

```
    odrl:assigner per:88 ;
    odrl:remedy [
      odrl:action odrl:anonymize ;
      odrl:target doc:77
    ] ;
    odrl:target doc:77
  ] .

## State of the world (historical)

per:88 odrlx:performsAction odrl:index ;
    odrlx:target doc:77 ;
    odrlx:timestamp "2025-03-05 17:26:49" .

per:87 odrlx:performsAction odrl:index ;
    odrlx:target doc:77 ;
    odrlx:timestamp "2025-03-06 08:42:32" .

per:88 odrlx:performsAction odrl:index ;
    odrlx:target doc:77 ;
    odrlx:timestamp "2025-03-06 09:05:08" .
```

Listing 3: Excerpt of sample Knowledge Graph containing an ODRL policy and the state of the world

Note that expressing the state of the world as a Knowledge Graph and linking the ODRL policies to it may require a certain ODRL ontology extension, as assumed in the example shown by Listing 3. In addition, in Section 3.6 authors will delve into the concept of policy variable that could be used as linkage between a policy and the state of the world.

3.3. Proposal 3: Status of the policy as a concept of ODRL

As different articles in the literature point out [11], in almost all scenarios, policies are activated and may or may not be fulfilled. These behaviours should be recorded to keep track of the historical life-cycle of one or more policies. In this sense, **authors propose to extend the ODRL ontology linking to each policy the concept of activation** holding different information relevant to know, such as whether the policy was fulfilled or not, who activated the rule, the timestamp of activation (and maybe the timestamp of the fulfilment). A draught of this extension is shown in Figure 2.

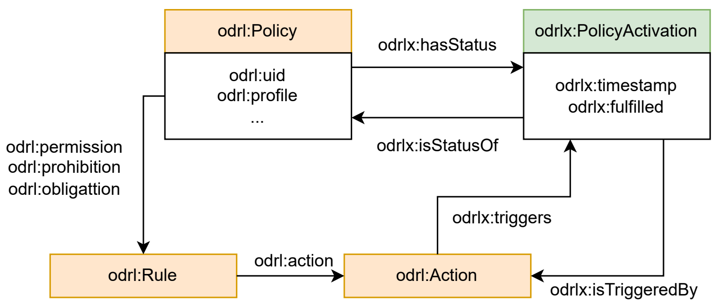

Figure 2: Draft of ODRL ontology extension for Activation

Note that the ontology described in Figure 2 is just a draught proposal and, in the case of adoption, the ODRL community will provide valuable feedback. A sample RDF excerpt using the activation term is shown in Listing 4, note that this excerpt is linked to the policy shown in Listing 3.

```
@prefix odrl: <http://www.w3.org/ns/odrl/2/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix odrlx: <http://www.w3.org/ns/odrl/2-extension/> .
@prefix : <http://www.example.org/activations/> .

:activation01 a odrlx:PolicyActivation ;
    odrlx:isStatusOf <http://example.com/policy:33CC> ;
    odrlx:fulfilled "true"^^xsd:boolean ;
    odrlx:timestamp "2025-03-05 17:26:49"^^xsd:datetime .

:activation02 a odrlx:PolicyActivation ;
    odrlx:isStatusOf <http://example.com/policy:33CC> ;
    odrlx:fulfilled "false"^^xsd:boolean ;
    odrlx:timestamp "2025-03-06 08:42:32"^^xsd:datetime .

:activation03 a odrlx:PolicyActivation ;
    odrlx:isStatusOf <http://example.com/policy:33CC> ;
    odrlx:fulfilled "true"^^xsd:boolean ;
    odrlx:timestamp "2025-03-06 09:05:08"^^xsd:datetime .
```

Listing 4: RDF excerpt showing a sample activation status for the policy shown in Listing 3

In numerous existing models of norms and policies [9, 10, 37, 12] it is common to consider that policies that are present in a system and have a precondition or activation condition that when fulfilled or satisfied makes the policy in force. For example, a prohibition to share a video may be in force in one specific country or in a specific time interval. The ability to obtain all active policies at a given instant of time is useful for an agent to plan its future actions by, for example, fulfilling obligations and avoiding violating prohibitions. It is also important in an open system where competing agents interact by exchanging digital resources to be able to detect whether a certain obligation or prohibition has been fulfilled or violated. This is crucial in order to be able to sanction bad behaviour or reward good behaviour and thus to have an expectation of the future behaviour of the agents with whom one interacts.

3.4. Proposal 4: Support various policy types

As discussed in Section 2.1, it would be crucial to extend the ODRL to allow it to express, both descriptively and in terms of the behaviour that a policy can prescribe, a wide range of different *types of policies*. In this article, we will illustrate some appealing types that have been found to be discussed in various models. It would, of course, be appropriate to collect use cases from those who are using ODRL in specific applications and to assess whether these cases fall within the various types of policy that have been identified (here or in other works) and possibly extend them.

When focussing on obligations, it is crucial to observe that currently in ODRL it is possible to express obligations governing the execution of an action carried out by a specific agent on the target of the policy. The list of the various types of actions that can be regulated is available in the ODRL vocabulary [38]. The obliged action can be further specified by using refinements. Such obligations may be conditional, i.e. only be in force when certain temporal or spatial conditions (to be assessed in the context in which the action is performed) are fulfilled.

There are at least two *types of obligations* that cannot be formalised in ODRL. The first is the obligation to perform an action before a specific deadline. Usually the exact value of the deadline is not known when the policy is defined but it changes at run-time and must be calculated based on when the policy is triggered by the occurrence of a specific type of event. For example, the obligation to pay a certain amount of money within a week of playing a certain film. For these types of obligations, when the deadline is passed without action being taken, there is a violation, which may involve a penalty, a penalty that may sometimes also require, when possible, meeting the initial obligation by a new deadline and paying a fine. The second type is the obligation to perform an action where the assignee is not specified at design time, but it becomes known when the obligation is activated at run-time. For example, a given service may decide to make the reading of the first chapter of a book available to all its users if they agree that once they have read the chapter, they will write a review on the chapter within a week of reading it.

To represent these types of obligations in ODRL, it is necessary to introduce into the ODRL ontology a specific class *Obligation* that is separate from the class of *Duty* as discussed in Section 2. It is also necessary to introduce the notion of *activation condition* of obligations (as aforementioned in Section 3.3), i.e. the description of a class of actions or of events, like in [10, 11, 12], and the notion of deadline. Furthermore, in order to be able to calculate the value of certain policy properties at runtime (for example the value of the actual deadline or the value of the actual assignee of an obligation), it is necessary to introduce the notion of *variables* as explained in 3.6.

As discussed in [39], considering that agents/parties are autonomous obligations, prohibitions, and permissions can be implemented by a system with two control mechanisms. One is represented by *regimentation* mechanisms that consist in making the violation of policies impossible. When focussing only on permissions, regimentation is called access control, i.e. when an action is not permitted, it is blocked. Regimentation may be difficult or impossible (e.g., regimenting an obligation to pay), or sometimes it is preferable to *allow agents to violate* the rules, and in this case the enforcement mechanisms are applied when violations are detected by reacting, for example, with *sanctions*. In this second approach, the meaning of a permission and therefore its evaluation mechanisms are different because violations are possible and other mechanisms are in place to discourage violations. For example, a patient may strictly decide that only doctors are allowed to read their data and prevents a nurse from doing the same. In a more flexible approach, a patient may decide that only doctors are allowed to read his or her data, and in the event of a breach of this prohibition (e.g. by a nurse), the latter must appear in front of a commission. **Due to aforementioned reasons, authors propose to introduce two different ontological classes into ODRL (or, in any case, the concept of regimentation) for the various permission concepts**, since such semantics are needed to differentiate different behaviours.

3.5. Proposal 5: SPARQL-based evaluation rules

Anytime the state of the world changes, policies should be evaluated in order to know if they are met or not. Another point of view is that whether the state of the world changes, policies must be evaluated to check if they are conformant to it. This mechanism is not defined in ODRL and is the main reason for researchers to have proposed numerous articles from the theoretical and practical point of view.

In this sense, and considering that ODRL is a W3C standard built on top of well-known standards like RDF or OWL, it is reasonable **to propose that ODRL adopts and promotes SPARQL-based evaluation rules** along with the proposal of introducing the concept of policy activation and fulfilment in their ontology (as discussed in Section 3.3). This proposal is also in alignment with the proposals from Section 3.1, since the policies will be in alignment with the SPARQL functions, and Section 3.2, since both the policies and the state of the world are expressed in RDF and thus can be queryable with SPARQL.

Assuming the Knowledge Graph from Listing 3 a set of SPARQL-based evaluation rules could be the ones shown in the following Listing 5.

```
PREFIX odrl: <http://www.w3.org/ns/odrl/2/>
PREFIX odrlx: <http://www.w3.org/ns/odrl/2-extension/>
PREFIX  : <http://www.example.org/activations/>
```

```
SELECT ?fulfilled ?result WHERE {

    # Policy
    <http://example.com/policy:33CC> odrl:prohibition ?prohibittion ;
        odrl:assigner ?allowedPerson ;
        odrl:target   ?targetResource .

    # State of the world
    ?person odrlx:performsAction odrl:index ;
            odrlx:target ?targetResource .

    BIND ( ?allowedPerson == ?person AS ?fulfilled )
    BIND ( if(?fulfilled, get(?targetResource), getAnonimized(?targetResource)) AS ?result )
}
```

Listing 5: Sample evaluation rule based on SPARQL SELECT query

Note that it is not required that SPARQL-based evaluation rules be necessarily SELECT queries. In fact, in Section 3.6, the possibility of using CONSTRUCT queries to generate a policy on the fly as a result of evaluating a different policy is explored. In addition, due to the proposal of Section 3.2 the SPARQL query could even be an update query [40] with results which are the activation triples mentioned in Section 3.3. Listing 6 shows an UPDATE SPARQL query which would update the Knowledge Graph shown in Listing 4 with new activation triples.

```
PREFIX odrl: <http://www.w3.org/ns/odrl/2/>
PREFIX odrlx: <http://www.w3.org/ns/odrl/2-extension/>
PREFIX fn : <http://www.w3.org/functions#>

INSERT {
    ?activationURI a odrlx:PolicyActivation ;
        odrlx:isStatusOf ?policy ;
        odrlx:fulfilled ?fulfilled ;
        odrlx:timestamp ?currentTimestamp  .
} WHERE {
    # Policy
    ?policy odrl:prohibition ?prohibittion ;
        odrl:assigner ?allowedPerson ;
        odrl:target   ?targetResource .
    VALUES ?policy { <http://example.com/policy:33CC> }

    # State of the world
    ?person odrlx:performsAction odrl:index ;
            odrlx:target ?targetResource .
    BIND ( ?allowedPerson == ?person AS ?fulfilled )
    BIND ( fn:createURI() AS ?activationURI )
    BIND ( fn:now() AS ?currentTimestamp )
}
```

Listing 6: Sample evaluation rule based on SPARQL UPDATE query

It is worth to mention that ODRL *actions* should be also ran within the SPARQL query (as shown in Listing 5). Also, from a conceptualisation point of view, the evaluation of a policy could be defined as a tuple $E_t = (P_{KG}, SoW_{KG}, Q)$ where P_{KG} is the policies Knowledge Graph (i.e., a set of RDF triples), SoW_{KG} is the state of the World Knowledge Graph (i.e., set of RDF triples), and Q is a set of SPARQL queries. This set of queries Q could contain similar queries as those shown in 5 or 6; These queries can leverage certain behaviours, such as performing the action of the policy (5) or automatically updating the policy activation status (6) and, in the case where both are contained in Q, a potential system could combine the benefits of both.

The main benefit of using SPARQL as evaluation rules is the fact that it is a W3C standard designed to work with RDF and OWL, which are the standards used in ODRL. In addition, SPARQL has the necessary expressiveness and formal semantics [41], to model the behaviour of policies that is independent of ODRL itself; as demonstrated in other articles [17]. Nevertheless, adopting SPARQL could bring some drawbacks, since it may require partitioners to have a deep knowledge of it, SPARQL may have limited reasoning capabilities in certain scenarios, and reduced performance in very large graphs.

3.6. Proposal 6: Policy templates & variables

As briefly discussed in Section 2.1, it is not possible to explicitly represent in ODRL policies that contain variables at design time that will be grounded to specific values when rules are evaluated at run-time (e.g., the policies discussed in items 1 and 2 of the next list). It is also impossible to represent policies that may generate many different specific obligations or prohibitions at run-time, such as the policies discussed in item 3 of the next list. These types of policies can be called: abstract (as in [12]), template, or pattern. For example:

1. A policy that has as recipients all agents of an interaction system who are unknown at the time the policy is designed, and therefore it is not possible to specify their identifiers as assignees of the policy.
2. A policy that contains an obligation with a deadline and the value of the deadline is computed when the obligation is activated by the execution of an action by an agent. For example, when a user reads a document, they must delete it from their system within two weeks.
3. A policy that contains an obligation or a prohibition that can be activated by the occurrence of an event or action belonging to certain classes. The properties of such events are necessary to compute the value of some properties of the obliged or prohibited action. For example, when someone reads a book, she/he is forbidden to read all other books in the same category in the following week. In this policy, the actor of the read action and the category of the book that was read are used to specify the properties of the subsequent specific prohibition.

We propose to introduce the notion of a policy template to refer to policies that contain variables and that, when activated, may generate more than one obligation or prohibition for various assignees. From the *Descriptive* perspective, a possible solution may consist of proposing a metalanguage over RDF to be able to specify variables as values of some of the properties of ODRL policies [17], for example, the assignee or the deadline. Another solution may consist in introducing the notion of a variable operator in the ODRL ontology. From the *behavioural* perspective, certain types of policies that contain variables will have to generate specific obligations or prohibitions when they are activated. This could be specified, for example, by using the CONSTRUCT clause of the SPARQL query language.

4. Conclusions

ODRL is steadily gaining traction in a broad range of contexts, such as European Data Spaces or Solid PODs, among others. Nevertheless, ODRL in its current version has severe limitations from the theoretical and practical point of view. This article presents a survey on the limitations and issues of ODRL reported in the literature and a taxonomy for categorising them. Based on such a hierarchy, in this paper have proposed specific ideas and proposals to improve these limitations and issues.

Through the analysis conducted in this article, it has been highlighted how ODRL can paves the way for diverse application scenarios, while also revealing key limitations related to its semantics (descriptive and behavioural) and the need for formalising new types of policies.

In future work, we aim to present these identified issues and proposed solutions to the ODRL Community Group. The goal is to assess the limitations and potentially evolve and integrate the proposed solutions. Furthermore, we plan to continue studying the solutions presented in this article with the aim of arriving at a more elaborate theoretical solution, creating frameworks, and evaluating the research results through case studies and practical experiments.

5. Acknowledgements

This work has been partially supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement 2023-2026 with the Universidad Politécnica de Madrid in Line A, Emerging PhD researchers through the project GUIA (M230020126A-AJCA). This work was conceived during the Dagstuhl Seminar 25051 *"Trust and Accountability in Knowledge Graph-Based AI for Self Determination"*, which took place in January 2025.

6. Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly and DeepL in order to: Grammar and spelling check, paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. Iannella, S. Villata, ODRL Information Model 2.2, Recommendation, W3C, 2018. https://www.w3.org/TR/2018/REC-odrl-model-20180215/.

[2] Z. Maamar, A. Benna, H. Kechaoui, ODRL-Based Provisioning of Thing Artifacts for IoT Applications., in: Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE, INSTICC, SciTePress, 2024, pp. 168–178. doi:10.5220/0012718600003687.

[3] A. Cimmino, J. Cano-Benito, e. M. García-Castro, Raúl", A. Skarmeta, S. Krco, A. González Vidal, The AURORAL Privacy Approach for Smart Communities Based on ODRL, in: Global Internet of Things and Edge Computing Summit, Springer Nature Switzerland, 2025, pp. 89–100.

[4] Sovereign-X, D1.3.1.2 Functional and Technical Architecture Specifications, Technical Report, European Commission, 2024. https://simpl-programme.ec.europa.eu/book-page/simpl-open-architecture.

[5] International Data Spaces Association, Technical Agreements, Technical Report, International Data Spaces Association, 2024. URL: https://docs.internationaldataspaces.org/ids-knowledgebase/idsa-rulebook/idsa-rulebook/4_technical_agreements.

[6] B. Esteves, H. J. Pandit, V. Rodríguez-Doncel, ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid, in: 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), volume , 2021, pp. 298–306.

[7] S. Steyskal, A. Polleres, Defining expressive access policies for linked data using the ODRL ontology 2.0, in: Proceedings of the 10th International Conference on Semantic Systems, 2014, pp. 20–23.

[8] M. Vacura, Modeling artificial agents' actions in context – a deontic cognitive event ontology, Appl. Ontol. 15 (2020) 493–527. URL: https://doi.org/10.3233/AO-200236. doi:10.3233/AO-200236.

[9] A. Garcia-Camino, P. Noriega, J. A. Rodriguez-Aguilar, Implementing norms in electronic institutions, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '05, Association for Computing Machinery, New York, NY, USA, 2005, p. 667–673. URL: https://doi.org/10.1145/1082473.1082575. doi:10.1145/1082473.1082575.

[10] M. Sensoy, T. J. Norman, W. W. Vasconcelos, K. Sycara, OWL-POLAR: A framework for semantic policy representation and reasoning, Journal of Web Semantics 12 (2012) 148–160.
[11] N. Fornara, M. Colombetti, Using Semantic Web technologies and production rules for reasoning on obligations, permissions, and prohibitions, AI Commun. 32 (2019) 319–334. URL: https://doi.org/10.3233/AIC-190617. doi:{10.3233/AIC-190617}.
[12] N. Fornara, S. Roshankish, M. Colombetti, A Framework for Automatic Monitoring of Norms that Regulate Time Constrained Actions, in: Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIV - International Workshop, COINE 2021, volume 13239 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 9–27. URL: https://doi.org/10.1007/978-3-031-16617-4_2.
[13] N. Fornara, V. Rodríguez-Doncel, B. Esteves, S. Steyskal, B. W. Smith, ODRL Formal Semantics, https://w3c.github.io/odrl/formal-semantics/, 2025. Accessed: (March 4, 2025).
[14] M. G. Kebede, G. Sileno, T. Van Engers, A Critical Reflection on ODRL, in: AI Approaches to the Complexity of Legal Systems XI-XII, Springer International Publishing, Cham, 2021, pp. 48–61.
[15] S. Steyskal, A. Polleres, Towards Formal Semantics for ODRL policies, in: Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings, volume 9202 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 360–375. URL: https://doi.org/10.1007/978-3-319-21542-6_23.
[16] A. Cimmino, J. Cano-Benito, R. García-Castro, Practical challenges of ODRL and potential courses of action, in: Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023, ACM, 2023, pp. 1428–1431. URL: https://doi.org/10.1145/3543873.3587628.
[17] A. Cimmino, J. Cano-Benito, R. García-Castro, Open Digital Rights Enforcement framework (ODRE): from descriptive to enforceable policies, Comput. Secur. 150 (2025) 104282. URL: https://doi.org/10.1016/j.cose.2024.104282.
[18] J. Polo, J. Prados, J. Delgado, Interoperability between ODRL and MPEG-21 REL., in: ODRL Workshop, Citeseer, 2004, pp. 65–76.
[19] J. Cano-Benito, A. Cimmino, R. García-Castro, Towards time privacy policies in ODRL, in: NeXt-generation Data Governance workshop 2024, 2024.
[20] M. De Vos, S. Kirrane, J. A. Padget, K. Satoh, ODRL policy modelling and compliance checking, in: Rules and Reasoning - Third International Joint Conference, RuleML+RR 2019, Bolzano, Italy, September 16-19, 2019, Proceedings, volume 11784 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 36–51. URL: https://doi.org/10.1007/978-3-030-31095-0_3.
[21] S. Roshankish, N. Fornara, How to Formalize Different Types of Norms in Multi-agent Systems: A methodology focused on the t-norm model, SN Comput. Sci. 5 (2024) 749. URL: https://doi.org/10.1007/s42979-024-03052-4. doi:{10.1007/S42979-024-03052-4}.
[22] R. García, R. Gil, I. Gallego, J. Delgado, Formalising ODRL semantics using web ontologies, in: Proc. 2nd Intl. ODRL Workshop, 2005, pp. 1–10.
[23] R. Pucella, V. Weissman, A formal foundation for ODRL, arXiv (2006).
[24] N. Fornara, M. Colombetti, Operational semantics of an extension of ODRL able to express obligations, in: Multi-Agent Systems and Agreement Technologies: 15th European Conference, EUMAS 2017, and 5th International Conference, AT 2017, Evry, France, December 14-15, 2017, Revised Selected Papers 15, Springer, 2018, pp. 172–186.
[25] I. Akaichi, W. Slabbinck, J. A. Rojas, C. Van Gheluwe, G. Bozzi, P. Colpaert, R. Verborgh, S. Kirrane, Interoperable and continuous usage control enforcement in dataspaces, in: The Second International Workshop on Semantics in Dataspaces, co-located with the Extended Semantic Web Conference, 2024.
[26] M. Lowry, T. Pressburger, G. Rosu, Certifying domain-specific policies, in: Proceedings 16th Annual International Conference on Automated Software Engineering (ASE 2001), IEEE, 2001, pp. 81–90.
[27] A. Hosseinzadeh, A. Eitel, C. Jung, A Systematic Approach toward Extracting Technically Enforceable Policies from Data Usage Control Requirements., in: ICISSP, 2020, pp. 397–405.

[28] L. Daniele, F. den Hartog, J. Roes, Created in close interaction with the industry: the smart appliances reference (SAREF) ontology, in: International Workshop Formal Ontologies Meet Industries, Springer, 2015, pp. 100–112.
[29] M. Poveda-Villalón, R. García-Castro, Extending the SAREF ontology for building devices and topology., in: LDAC, 2018, pp. 16–23.
[30] B. Esteves, V. Rodríguez-Doncel, H. J. Pandit, N. Mondada, P. McBennett, Using the ODRL Profile for Access Control for Solid Pod Resource Governance, in: The Semantic Web: ESWC 2022 Satellite Events, Springer International Publishing, Cham, 2022, pp. 16–20.
[31] O. Corcho, M. Fernández-López, A. Gómez-Pérez, Methodologies, tools and languages for building ontologies. where is their meeting point?, Data & knowledge engineering 46 (2003) 41–64.
[32] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The NeOn methodology for ontology engineering, in: Ontology engineering in a networked world, Springer, 2011, pp. 9–34.
[33] R. Albertoni, D. Browning, S. J. D. Cox, A. Gonzalez-Beltran, A. Perego, P. Winstanley, Data Catalog Vocabulary (DCAT) - Version 3, in: W3C Recommendation, 2024, p. .
[34] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, et al., Brick: Towards a unified metadata schema for buildings, in: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, 2016, pp. 41–50.
[35] A. Bröring, C. Stasch, J. Echterhoff, OGC Sensor Observation Service Interface Standard, Version 2.0. (2012).
[36] B. De Meester, A. Dimou, R. Verborgh, E. Mannens, An ontology to semantically declare and describe functions, in: The Semantic Web: ESWC Satellite Events, Heraklion, Crete, Greece, May 29–June 2, 2016, Revised Selected Papers 13, Springer, 2016, pp. 46–49.
[37] T. Athan, G. Governatori, M. Palmirani, A. Paschke, A. Z. Wyner, LegalRuleML: Design Principles and Foundations, in: Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures, volume 9203 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 151–188. URL: https://doi.org/10.1007/978-3-319-21768-0_6.
[38] R. Iannella, M. Steidl, S. Myles, V. Rodríguez-Doncel, ODRL Vocabulary & Expression 2.2, Recommendation, W3C, 2018. https://www.w3.org/TR/2018/REC-odrl-vocab-20180215/. Latest version available at https://www.w3.org/TR/odrl-vocab/.
[39] N. Criado, E. Argente, P. Noriega, V. Botti, MaNEA: A distributed architecture for enforcing norms in open MAS, Eng. Appl. Artif. Intell. 26 (2013) 76–95. URL: https://doi.org/10.1016/j.engappai.2012.08.007.
[40] P. Gearon, A. Passant, A. Polleres, SPARQL 1.1 Update, in: W3C Recommendation, 2013, p. .
[41] J. Pérez, M. Arenas, C. Gutierrez, Semantics and Complexity of SPARQL, in: International semantic web conference, Springer, 2006, pp. 30–43.