

Towards ODRL 3.0

A use-case based analysis of ODRL 2.2

Wouter Termont^{1,*}, Wout Slabbinck¹, Joachim Van Herwegen¹, Patrick Hochstenbach¹ and Beatriz Esteves^{1,*}

¹Knowledge On Web Scale, IDLab, Ghent University – imec, Technologiepark-Zwijnaarde 122, 9052, Ghent, BE, <https://knows.idlab.ugent.be/>

Abstract

The increasing use of automated, machine-readable policy systems for governing digital content and data usage has made robust and interoperable policy languages increasingly important. The Open Digital Rights Language (ODRL) aims to support such use cases but faces practical limitations in real-world applications. ODRL 2.2 lacks formal semantics and exhibits ambiguities in its conceptual model, limiting its scalability, interoperability, and reliable automated evaluation. This work analyzes ODRL 2.2 through a music distribution use case to identify conceptual and operational shortcomings. It examines limitations in ODRL’s conceptual model – such as collections, actions, and extensibility – as well as operational aspects including constraints, evaluation, and the representation of the state of the world, and proposes potential improvements. Overall, the article concludes that a formal semantic foundation and clearer model are essential for ODRL to enable consistent and interoperable policy evaluation. Future work toward ODRL 3.0 should therefore focus on formal semantics, simplified abstractions, and standardized evaluation mechanisms to support scalable policy-driven systems.

Keywords

ODRL, policy, digital rights, usage control, authorization, semantics, formalization

1. Introduction

The Open Digital Rights Language (ODRL) is a *policy expression language*: it allows one to express usage control policies – normative rules governing actions on resources – in a machine-readable way, to support their transparent, systematic and automatic application, thereby “[enabling] parties who hold certain rights on [resources] to exercise those rights” [1].¹ The ODRL Information Model is a descriptive semantic model, i.e., a logical view of the necessary components, while the ODRL Vocabulary & Expression specification provides an normative, implementable view of the model [3, 4].

However, actual application – beyond mere (de)serialization – has proven difficult. Research and implementation experiences with ODRL 2.2 have uncovered a number of problems with its current specifications. We distinguish two kinds of issues: **conceptual** ones, including terminological inconsistencies, ambiguities, redundancy, and arbitrariness, as well as restrictions on the **extensibility** of the model; and **operational** ones, in particular the lack of a formal semantics. In this paper, we explore these conceptual limits (in Section 2) and operational challenges (in Section 3), and exemplify them with several scenarios around a use case involving music distribution. For each issue, we refer to related work, and suggest different approaches that could address them in a future version of ODRL.

The use case guiding us through each of the issues revolves around DistriNet, a music distribution network, and FancyBand, one of their registered artists. We attempt to model several scenarios

ODRL and beyond: practical applications and challenges for policy-base access and usage control. OPAL 2026 Co-located with the Extended Semantic Web Conference, Dubrovnik, Croatia, May 10-11, 2026

*Corresponding author.

*Corresponding author.

✉ wouter.termont@ugent.be (W. Termont); wout.slabbinck@ugent.be (W. Slabbinck); joachim.vanherwegen@ugent.be (J. V. Herwegen); patrick.hochstenbach@ugent.be (P. Hochstenbach); beatriz.esteves@ugent.be (B. Esteves)

🆔 0000-0002-2968-1394 (W. Termont); 0000-0002-3287-7312 (W. Slabbinck); 0000-0002-8215-1827 (J. V. Herwegen); 0000-0001-8390-6171 (P. Hochstenbach); 0000-0003-0259-7560 (B. Esteves)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹Other, less expressive policy languages include the Extensible Access Control Markup Language (XACML) and the Enterprise Privacy Authorization Language (EPAL) [2].

involving their policies around the distribution and derivation of works. Examples are provided as RDF 1.1 in TriG notation.² For the RDF vocabulary used throughout this paper, we predominantly use the relative dummy namespace # (prefix :), to avoid unnecessary assumptions, and to emphasize ODRL's extensibility. All other prefixes are listed in the ODRL Vocabulary [4, 2.1].]

2. Conceptual limitations

A central characteristic of robust Web standards is a simple conceptual model that provides a domain-agnostic, extensible abstraction of a recurring pattern. This makes a specification applicable to many different domains that contain a concrete instantiation of the pattern. It also facilitates the reuse of the model in other specifications, as well as a backward-compatible evolution over time. Domain-specific differences and other variations can ideally be plugged into foreseen extension points of the vocabulary or the protocol.

The ODRL Community Group (ODRL CG) and the Permissions & Obligations Expression Working Group (POE WG) of the World Wide Web Consortium (W3C) definitely took this too heart when designing the specification. Its core abstraction are **rules** (bundled into *policies*): deontic modes over operations, consisting of *actions* performed by *parties* on *assets*. While this is a relatively simple, abstract conceptual model, the expressiveness of each of its components is crucial to be widely applicable. To keep these representations flexible, ODRL includes several hierarchically extensible structures. In particular, hierarchies of parties and assets can be modelled as *collections* (with `odr1:partOf`), while actions can be encompassed by other actions (with `odr1:includedIn`). A rule that applies to the object of such a relation automatically applies to all subjects.

Despite this abstract, extensible model, some design choices of the specification nevertheless severely limit its final expressiveness. In the following subsections, we will highlight the limitations introduced by certain details of the hierarchical structures, and point out two issues with the extensibility and reusability of the specification.

2.1. Collections

ODRL collections provide an elegant way to apply a rule to multiple assets or parties at once. Asset collections function as assets themselves, and party collections as parties. Rules containing a collection hold for every member related to it via the `odr1:partOf` property (unless a refinement is applied, cf. Section 3.1). In the case of asset collections, however, this leads to two undesirable effects:

- Since a rule that targets an asset collection is interpreted as applying to each of its members, it is not possible to define a rule that holds on the asset collection itself.
- When mapping a domain to ODRL in which more than one hierarchical relationship holds between the same entities, these all map to `odr1:partOf`, thereby losing crucial information to distinguishing between them.

We exemplify these issues in Scenario 1, which Listing 1.1 represents as a basic domain model using a custom vocabulary. A straightforward mapping to ODRL 2.2 then results in Listing 1.2, where the issues become apparent:

- Since any rule targeting the album applies to its tracks, it is not possible to define policies that apply to the album itself. We could potentially interpret such rules as *also* holding for the album itself; but then albums would be subjected to exactly the same rules as their tracks, and vice versa – which will often not be desired.
- Without any additional information, the `:remixOf` and `:coverOf` relations between `FancyRemix` and `FancySong` are both mapped to `odr1:partOf` membership relations of asset collections that cannot be distinguished from each other. While we could potentially resolve this by introducing

²TriG is an extension of Turtle, extending it with a syntax to express RDF graphs.

custom subclasses or labels – e.g., as part of a music-specific ODRL extension profile – such an approach necessarily depends on the available domain-specific information, and thus precludes a uniform – automatable – mechanism.

In the *Authorization for Data Spaces* specification [5], we suggest to solve these issues by modelling collections as ‘**proxy assets**’, distinguished by the combination of a ‘proper’ *source asset* (indicated with `odr1:source`) and a corresponding *domain-specific member-relationship* (indicated with `:relation` and `:inverseRelation` in the example),³ which conveys the particular meaning of the `odr1:partOf` statements of each collection.

The rationale behind this proposal is a threefold increase in the model’s expressivity:

1. Rules on the source asset apply (only) to itself, while those on the proxy asset apply to all of its members – and not to the source asset.
2. A single asset can be the source for multiple proxy assets, representing different ways in which it might form a collection of other assets.
3. The concrete member-relationship of a domain is the only data point that is truly necessarily to derive an `odr1:partOf` hierarchy; and thus the only foundation for a uniform mechanism. Tracking this relationship as part of the proxy asset’s metadata not only allows us to uniformly distinguish between collections; it is also the ideal starting point for discovering additional information about the nature of each collection, e.g., their domain, range, labels, and profile-specific properties.

When we apply this to Scenario 1, it can be modelled as in Listing 1.3.

2.2. Actions

In ODRL, actions express the different operations that can be performed on an asset, and to which rules can ultimately apply. While the specification leaves it open to external vocabularies and extension profiles to come up with domain-specific action terms, it nevertheless limits this extensibility in two ways:

- Each action must (transitively) be included in one of two *top-level actions*: `odr1:use` (using the asset) or `odr1:transfer` (transferring ownership of the asset). Several authors have already remarked how this complicates the expression of actions related to other life cycle events (creation, publication, deletion), different kinds of transfers (e.g., physical movement, stewardship, delegation), or the difference between primary and secondary use [1, 6, (#35, #96, #112, #140)].⁴
- It is not possible to represent *compound actions*. While ODRL’s *rule composition* mechanism allows one to write *compound rules* with more than one action, this is merely syntactic sugar: they are evaluated as the conjunction of several separate (atomic) rules with exactly one action.

Scenario 2 exemplifies these issues with the action model. The basic ODRL 2.2 representation in Listing 2.1 shows how it can result in confusing, counter-intuitive semantics. It is unclear what the nature and target of the `:create` action are; and the compound prohibition normalizes into two separate rules – neither of which correctly convey the desired semantics.

Again, both limitations could potentially be solved together, by generalizing the object of a rule. While ODRL is primarily aimed at *agentic operations* – involving a responsible agent – we could instead interpret actions as a subclass of the more abstract notion of *events*. By relying on an event model, one

³The purpose of `:inverseRelation` is merely to facilitate referencing membership relations that are defined in the opposite direction of `odr1:partOf`. Instead of introducing two separate predicates for this, one could also opt for a blank node with `owl:inverseOf`. For example: `_:collection :relation [owl:inverseOf :track]`. Note that this is purely descriptive, so OWL reasoning is not relevant here.

⁴The ODRL editors have acknowledged this limitation, and concrete steps have already been taken to solve it; e.g., the introduction of `odr1:create` in the ODRL Community Vocabulary. Nevertheless, a truly extensible solution should address this issue in the model itself, instead of merely extending the arbitrary list of top-level actions (cf. Section 2.3).

Scenario 1. *To regulate the fair distribution of earnings towards registered artists, DistriNet wants to formulate policy rules over individual recordings and whole albums, as well as several types of derived work. Amongst these are covers of other artists performing a song, and remixes which reuse (part of) the original recording. A particular case is a recording called FancyRemix, which is both a remix and a cover of the popular track FancySong of the album FancyAlbum.*

Listing 1.1 A domain model of Scenario 1

```
:fancy-album
  :track :fancy-song .
:fancy-remix
  :remixOf :fancy-song ;
  :coverOf :fancy-song .
```

Listing 1.2 ODRL 2.2 mapping of Scenario 1

```
:fancy-album a odrl:AssetCollection .
:fancy-song a odrl:Asset ;
  odrl:partOf :fancy-album .
:fancy-remix a odrl:Asset ;
  odrl:partOf [
    # FancySong covers
    a odrl:AssetCollection
  ] , [
    # FancySong remixes
    a odrl:AssetCollection
  ] .
```

Listing 1.3 Alternative mapping of Scenario 1

```
:fancy-album a odrl:Asset .
:fancy-song odrl:partOf [
  a odrl:AssetCollection ;
  odrl:source :fancy-album ;
  :inverseRelation :track
] .
:fancy-remix odrl:partOf [
  a odrl:AssetCollection ;
  odrl:source fancy-song ;
  :relation :remixOf
] , [
  a odrl:AssetCollection ;
  odrl:source fancy-song ;
  :relation :coverOf
] .
```

could model ‘actions’ that do not require an target identifier, or ones that are a compound of multiple ‘sub-actions’.⁵

In Listing 2.2, we sketch how such an approach could express Scenario 2, using the Simple Event Model [9].⁶ A comparison with other event models [e.g., 10] is out of scope for this paper.

2.3. Extensibility & reuse

Apart from the structural problems related to the hierarchies of parties, assets, and actions, a few recurring tendencies of the ODRL specification additionally hinder its extensibility and reusability.

First, the ODRL vocabulary introduces several terms that take up a function already fulfilled by terms from other standard vocabularies. These include the use of `odrl:uid` to indicate identifiers of several components (policies, rules, assets, parties, constraints ...), instead of their RDF node identifier (IRI) – already raised in [6, (#26, #30)] – and the introduction of `odrl:partOf` in favor of the semantically identical `dc:isPartOf`, part of the popular Dublin Core vocabulary. Such redundant duplication of already standardized semantics is confusing towards developers that are unfamiliar with ODRL – potentially withholding them to take the step to use it – and diminishes the reusability of ODRL in relation to specifications that rely on these conventions.

Second, and related to the previous point, in its Vocabulary & Expression [4], ODRL introduces a ‘common vocabulary’ of ‘generic terms’, meant to be used in extension profiles. It is an arbitrary list

⁵Interestingly, such an event-based approach to ODRL has also been explored as a way to represent the temporal aspect of actions [cf. 7, 8].

⁶Using prefix `sem:` for the namespace (<http://semanticweb.cs.vu.nl/2009/11/sem/>) [9].

Scenario 2. *DistriNet wants to be able to formulate policies governing the life-cycle of recordings (e.g., creation). Moreover, since distribution of earnings quickly got complicated, they want to prohibit artists deriving a single recording in multiple ways (e.g., remixing and covering).*

Listing 2.1 Attempted ODRL 2.2 model of Scenario 2

```
:remix a odrl:Action ;
  odrl:includedIn odrl:use .
:cover a odrl:Action ;
  odrl:includedIn odrl:use .
:create a odrl:Action .
  # odrl:includedIn ... ???

_:p1234 a odrl:Policy ;
  odrl:permission [
    odrl:action :create
    # odrl:target ... ???
  ] ;
  odrl:prohibition [
    odrl:target :dn-records ;
    odrl:action :remix , :cover
  ] .
# === Normalization ===
# odrl:prohibition [
#   odrl:target :dn-records ;
#   odrl:action :remix
# ] ;
# odrl:prohibition [
#   odrl:target :dn-records ;
#   odrl:action :cover
# ] .
```

Listing 2.2 Alternative model of Scenario 2

```
# === No action hierarchies required ===

_:p1234
  odrl:permission [
    # A targetless event
    odrl:action :create
  ] ;
  odrl:prohibition [
    # A compound event
    odrl:target :dn-records ;
    odrl:action [
      sem:hasSubEvent :remix , :cover
    ]
  ] .
```

of terms, covering an unspecified choice of domains, with little consistency in the structure of their definitions – mostly formulated as prose ‘notes’. In Section 3.2, we come back to this lack of consistent and machine-readable semantics. Here, we will just point out the infelicity of such terms in a standard.

In general, robust standards refrain from touching upon any aspect that can be specified orthogonally to it – i.e., which can be modularly combined without requiring additional integration. This includes semantics that can be used from other standards without modifications, as well as any domain-specific applications.⁷ The potential of a standard depends just as much on what it does not define as on what it does. Key factors in extensibility and reuse are sharply delineated extension points wherever orthogonal modularity makes sense, and a way to declare and discover which extensions a specific application supports – neither of which is provided by ODRL 2.2 [6, (#69)].

3. Operational challenges

In the previous section we discussed several issues with ODRL conceptual model, which limit its semantic expressivity. Up until ODRL 2.2, the semantics of the language have always been purely *descriptive*: defining the policies that can be serialized in the language – and subsequently deserialized

⁷It also applies to many of the features proposed on the ODRL GitHub repository, e.g., the introduction of an ODRL-specific DID method [6, (#101, #102)], an ODRL URI scheme [6, (#131)], or a policy inheritance mechanism [6, (#63)].

from it. While this suffices to transfer the conceptual *content* of policies in a standardized way, it fails to convey their desired *effects*. This makes it hard to operationalize ODRL policies in actual applications, which often need to ensure the consistent evaluation and enforcement of policy rules. In the following subsections, we therefore look into a number of operational challenges with ODRL 2.2, and propose several ways these could be dealt with.

3.1. Constraints

Central to the expressiveness of ODRL policies are constraints: expressions that limit the applicability of policy rules, either as a condition on the rule itself (`odr1:constraint`), or as a refinement of its party (collection), asset (collection), or action (`odr1:refinement`) [3]. Each constraint consists of a comparison *operator* and two *operands* to compare: a variable (the **left** operand), and a value to compare it to (the **right** operand) – or an IRI reference to it [3].⁸

While this allows for a lot of freedom and extensibility, certain design choices make it difficult to operationalize the model in practical use cases. Some of these have already been addressed in earlier work, such as the lack of information to deterministically obtain a value from a *right operand reference* [11].⁹ In this section, we focus on the lack of operational semantics of the **left operand**, and the difference between *constraints* and *refinements*.

Definitions of left operands are given in natural language – often merely a few sentences. A quick look at the operands defined in the ODRL Vocabulary should make it clear that even for humans it is hard to grasp the variety of ways in which operands relate to policies and the state of the world [4]. For a machine, this is impossible – especially in a consistent and scalable manner.¹⁰

Moreover, it is often unclear whether a constraint should be modelled as a *refinement* of the action, of the asset, of a party (`odr1:refinement`), or as a condition on the rule itself (`odr1:constraint`) [cf. 6, (#73)]. This ambiguity has been repeatedly discussed in the ODRL CG. The editors have clarified that constraints on a rule are meant to “describe the [high-level] world state under which [it] becomes effective,” while refinements on actions “narrow the [intrinsic] semantics of the action operation directly,” and refinements on collections “indicate the context under which to identify individual [members] of the complete collection” [6, (#60)]. Supposedly, this distinction is meant to increase flexibility in expressing policies; though it is unclear what such flexibility would consist of – especially considering the amount of ambiguity it introduces.

Consider Scenario 3.¹¹ Listing 3.1 formalizes the desired constraint using the left operand `odr1:spatial`. The ODRL Vocabulary defines this operand as “a named and identified geospatial area with defined borders” [4]. However, neither the exact semantics nor the benefits of each option are clear:

- As a refinement on the **action** (Listing 3.2), the constraint restricts the prohibition to ‘**download operations in some way related to the geospatial area of the US**’. While a human interpreters might have some vague intuitions about what this entails, it does not suffice to apply the policy rule in an operational setting. Does it mean that the device on which the song is downloaded cannot physically be on US territory; that the download cannot transfer via an internet provider registered in the US; that it cannot originate from a server situated in the US ...?
- As a refinement on the **assignee** (Listing 3.3), the constraint restricts the prohibition to ‘**subscribers in some way related to the geospatial area of the US**’. Again, apart from intuition, the

⁸A constraint may optionally specify the *data type* and *unit* of the values under comparison, as well as a *status* – tracking the value of the left operand variable for the next comparison. Multiple constraints can be combined into *logical constraints* with a boolean operator [3].

⁹The solution proposed by the authors is to add a (SHACL) property path, which can be followed to obtain the value from an RDF representation of the right operand reference [11]. An implementation of this proposal is provided in [12]

¹⁰Several instances of this more general problem have already been brought to the attention since the publication of ODRL 2.2. One paper contrasts the scalar (i.e., one-dimensional) nature of the model to the multi-dimensional (vector) semantics of several left operands [13]; and an issue of the ODRL Community Group highlights ambiguity of left operands in refinements of collections that can have other collections as members (e.g., DCAT catalogs, series, and datasets) [6, (#108)].

¹¹This scenario is inspired by the examples in [6, (#145)].

Scenario 3. *In negotiating with DistriNet about the distribution of their work, FancyBand wants to specify that their songs cannot be downloaded in the U.S.*

Listing 3.1 An ODRL 2.2 spatial constraint ...

```
:us-constraint a odrl:Constraint ;
  odrl:leftOperand odrl:spatial ;
  odrl:operator odrl:eq ;
  odrl:rightOperand :US .
```

Listing 3.2 ... as action refinement

```
_:p2345 odrl:prohibition [
  odrl:action [
    rdf:value :download ;
    odrl:refinement :us-constraint
  ] # ...
] .
```

Listing 3.3 ... as party refinement

```
_:p2345 odrl:prohibition [
  odrl:assignee [
    odrl:source :distrinet-users ;
    odrl:refinement :us-constraint
  ] # ...
] .
```

Listing 3.4 ... as rule condition

```
_:p2345 odrl:prohibition [
  odrl:constraint :us-constraint
  # ...
] .
```

operational effects of the rule are unclear. Does it mean that the song cannot be downloaded by US citizens; by anyone located in the US at the start of the download, for the duration of the download, or at the time the policy is being evaluated?

- As a condition on the **rule** itself (Listing 3.4), the constraint restricts the **application** of the prohibition *in some way to the geospatial area of the US*. Since this **application** is explicitly distinguished from refinements – which are, perhaps indirectly, also about the rule’s application – it is unclear what such a constraint would mean. After all, any constraint is ultimately about some specific aspect of a rule. Does a condition on a rule merely function as catch-all for aspects that are not one of the main ODRL roles (asset, party, action)? If not, it perhaps restrict the life cycle of the rule itself; then the example constraint could mean that it cannot be proposed, agreed, evaluated... on US soil.

The scenario illustrated above should convey a good sense of the operational challenges involved in interpreting the different ways in which a single constraint can be applied to in a policy rule. Note that the constraint in question is actually still relatively simple: it pertains to the relation between some aspect of the rule and a fixed geographic indication. The challenges increase even more when also considering the infinite possible constraints on temporal aspects, purpose and legal grounds, processing restrictions, data retention, (sub)delegation ...

Moreover, since the semantics of the constraint’s left operand (`odrl:spatial`) are not expressed in a formal way, it is unclear how to obtain a concrete value for it in any particular situation – from a request, file metadata, a database, etc. Neither is it clear how to process such a value. This holds even in the case of `odrl:spatial`, which explicitly specifies that it must be an IRI that refers to “a *code value* for the area and [the] *source* of the code” [4]. Should one dereference the IRI, or look it up in a registry; will the result be RDF, or another format; will the country code format be encoded as datatype or as a property, and which one ...? It is impossible to implement a sound evaluation algorithm that can consistently process this rule as a list of actionable checks.

The ODRL editors seem to rely on the ‘operational community’ to define left operand semantics using extension profiles [6, (#115, #144)]. However, such an approach is necessarily limited to specific domains of application [e.g., 14, 15, 13], of which it requires policy authors and evaluator implementers to have a precise structural knowledge. This is not scalable, and risks to end up in a fragmented

Scenario 4. Because FancyBand gets a lot of requests from artists who would like to create a remix or cover of their popular FancySong, they want to let an automated agent handle these, based on some preset rules.

landscape of domain-specific vocabularies processed by non-interoperable evaluators [cf. 6, (#111)].

To define and evaluate left operands in a scalable, extensible way, ODRL itself should instead require ontology editors to provide the necessary information – the exact subject of the operand, as well as the source and format of its actual value – in a uniform, structured way, and define a general algorithm that can process this structure. Interesting options that have already been proposed in the past include a standardized operand *signature* (cardinality, data types, and property paths) [6, (#111)], or the use of external specifications that provide an *evaluation model* (e.g., the Function Ontology (FnO) [16]).

Once the precise operational details of a constraint’s left operand are known, it also becomes clear that the location of constraints – as refinement on actions or parties, or as condition on the rule – loses all significance. After all, the operational semantics will be fully understood from interpreting the constraint itself, regardless of where it is situated. From a purely conceptual perspective, it would therefore make sense to define all constraints as “refinements on (the execution of) the action”, since this is what the deontic mode of the rule is ultimately about. [Interestingly, the same can be said about the target(s) and assignee(s) themselves: a permission, obligation, or prohibition applies to the action being executed, *conditional upon* the execution being done on a specific asset by a specific party. Not only *refinements* on targets and assignees, but also the target and assignee aspects themselves can thus semantically be considered as constraints on (the action of) a rule.]

3.2. Evaluation

As mentioned in the introduction, the main goal of having a machine-readable policy expression language like ODRL is to support the transparent, systematic and automatic application of usage control policies. However, while the ODRL Vocabulary [4] indeed allows us to codify such policies according to the conceptual ODRL Model [3], neither document specifies the *operational semantics* – the intended *effect* of policies. Because of this, implementing actual applications around these representations has proven difficult.

To address this, the W3C ODRL CG has started to draft an ODRL Formal Semantics specification, which defines the (expected) behaviour of an *ODRL Evaluator* [17]. This software component should be able to determine within a given context **(a)** which rules are in effect, **(b)** whether a certain action is permitted, and **(c)** whether an obligation has been fulfilled (or prohibition violated) [17]. However, the draft once again merely *describes* – be it with more detail and examples – what the different states of evaluation are, rather than actually formalizing *how* exactly these are to be executed in practice.

Moreover, the list of three objectives is quite limited to provide a comprehensive formalization of all the ways in which policies can be used. To contrast, [18] emphasizes the need to formalize *seven* processes: validation, normalisation, conversion, authorization, monitoring, profiling, and conflict resolution. Scenario 4 sketches a simple requirement pertaining to our use case, which includes at least the following functions:

- **Relevance check:** determining which rules are *relevant* for a situation (e.g., a request), or fall back to a (configurable) default policy [cf. 6, (#123), and (a)].
- **Consistency check:** checking whether multiple rule sets are *consistent* (e.g., a request and policy) [cf. (b)].
- **Conflict resolution:** resolving *conflicts* between inconsistent rule sets (e.g., a request and a policy), taking into account conflict strategies [cf. 19, 20], as well as contextual information [cf. 1]; for example, by constructing an *alternative* rule set (e.g., a closest counter-offer) [cf. 6, (#110)], or extracting the right questions to *defer* the decision to a human-in-the-loop.
- **Policy instantiation:** constructing concrete *instantiations* of given rule sets (e.g., an offer based on a request and a policy).

- **Audits:** determining how a (later) state of the world relates to a rule set (e.g., fulfilment of obligations, violation of prohibitions) [cf. (c)].

While the Formal Semantics draft sketches a *signature* for an evaluator that can handle some of these objectives, and describes its *behavior* with examples [17], it lacks a concrete *algorithm* to get from its input to its output. Multiple external proposals exist providing a mathematically rigorous algorithm to reason *about* policy states [19, 8], or suggesting a specific implementation (e.g., production rule systems [7], answer set programming [14], or SPARQL queries [16]). However, these approaches do not address the core issue raised in the previous section: they still consider only a *limited domain*, modelling each variable separately with custom constraint semantics. As stated before, without a uniform mechanism to formalize them, this is not scalable.

Nevertheless, the formalizations mentioned above constitute an important contribution towards a more unified approach: by describing policies mathematically as objects in a (partially) ordered space, they emphasize a *comparative semantics* [cf. 19, 8]. This kind of formal semantics describes policies relative to each other, determining the inclusion of one policy in another, and measuring the distance between them based on their differences and similarities. One can see this inclusion relation as an *instantiation* (or ‘*implication*’) from more abstract policies to more concrete ones [8] – in which the former serves as a more generic ‘*template*’ for the latter [6, (#65); 16].

However, while both papers stop at the formalization of their proposed comparative semantics [19, 8], two even stronger insight follows from it:

1. Since all policies – even partial ones – can be situated as comparable objects in a (partially) ordered mathematical space, the difference between the several operational objectives cannot be found on an *ontological* level – e.g., based on different types of actions, rules, or rule sets [cf. 17, 6, (#69, #74, #90, #93)]. Rather, the distinction between the different evaluation processes is situated on a *functional* level, i.e., how we *use* that one generic, comparable kind of policies.
2. While the distinct processes might seem largely independent, even on a functional level there lies a strong parallel at their core: they all *compare* one or more of those objects – (partial) rule sets – determining how one relates to another, either in isolation or in the context of a state of the world. Relevance checks compare assets, actions, and parties; consistency checks and audits compare constraint operands; instantiation calculates an *order*; conflict resolution calculates a *distance*. Therefore, while the term ‘*evaluation*’ is ambiguous – referring to a number of independent objectives – it always entails a comparison of rule sets and states of the world. Each different form of evaluation could therefore potentially be grafted upon a single shared *comparison algorithm*.

Without going deeper into the practicalities of such an approach, we limit ourselves here to emphasizing its strong potential as a uniform foundation for a formal, operational semantics of ODRL.

3.3. State of the World

The main challenge, for making such a shared algorithm feasible, is how to express its input and output (messages) in a uniform way. In particular, the question remains how to represent a state of the world – sufficiently corresponding to reality – so that an evaluator can process it (effectively), and compare aspects of it to (parts of) the given policies.

As emphasized in the ODRL Community Group, “before discussing *how to represent* information in the state of the world ... it is best to reach an agreement on *what to represent*” [6, (#61)]. Parameters that have already been proposed include the sets of assets and parties (and their collections); the current time; and the list of performed actions [6, (#61)]. More generally, it should include all the additional information that is required at the time of evaluation, but not provided by the active policies [6, (#113)]. Potential sources of this information include “the system that is evaluating the policy (e.g., the current date and time), a requester (who may provide certain data needed in the policy), or an external system” [6, (#68)].

In this context, [21] introduced a *state of the world (SotW) model* to represent contextual facts that are needed to evaluate data-usage policies written in ODRL. This work formalizes relevant conditions (temporal, spatial, payments, or other contextual factors) as structured knowledge that an ODRL evaluator can use when deciding whether a requested action is permitted, prohibited or obligatory.

Moreover, discussions on how to represent and acquire state of the world contextual information are not unique to ODRL-based systems. Other policy-based systems have run into similar issues. The eXtensible Access Control Markup Language (XACML) [22], an OASIS standard which describes a syntax for an access control policy language and functional requirements for its evaluation and enforcement, provides a few solutions to this issue via its Policy Information Point (PIP) component, i.e., “system entity that acts as a source of attribute values”. In this context, the PIP component is in charge of acquiring the necessary attributes for policy evaluation, which can be retrieved from multiple sources, i.e., resources, subjects, environments. One can easily argue that ODRL-based systems should adopt a similar strategy. The modelling of such attributes is however limited to only an abstract class, i.e., *Attribute*. NIST SP 800-162 [23], a guidance document by the National Institute of Standards and Technology (NIST) on Attribute Based Access Control (ABAC), suggests a similar approach by providing additional considerations for *environment conditions*, i.e., “dynamic factors, independent of subject and object, that may be used as attributes at decision time to influence an access decision”, which should be managed by a PIP via “some sort of context handler or workflow coordinator”. Again, the modelling of such conditions is left out of scope.

Next to a uniform formalization of the algorithm’s input, it should be possible to represent its output, so it can be reliably used in other processes (e.g., resolution), integrated into the state of the world (e.g., proofs of audit results) [18].¹² One proposal for such output representation is the Compliance Report Model, introduced in [24], which captures which rules are in effect, and allows one to express the state of duties. It is part of the Framework for ODRL Rule Compliance through Evaluation (FORCE) [25], which also includes a test suite¹³ and a reference implementation of a compliant evaluator.

By formalizing inputs and corresponding outputs, the contributions mentioned in this section define the correct behaviour of an ODRL evaluator, and ensure inter-engine interoperability within the ODRL ecosystem.

4. Conclusion

Summarizing the conclusions of the previous sections, a next version of ODRL should address the limitations of the current conceptual model (cf. Section 2), and the challenges with its operationalization (cf. Section 3).

Proposed changes towards a more flexible, less ambiguous *conceptual model* include

- a more generic action model (cf. Section 2.2);
- a more generic approach to asset collections (cf. Section 2.1); and
- a critically evaluation of the specification’s reusability, as well as its own reuse of other standards pertaining to orthogonal concerns (cf. Section 2.3).

The operational challenges, on the other hand, can be addressed by the addition of a *formal semantics*, which should include

- a operational simplification of the constraint model – possibly around the notion of refinements on events – uniformly indicating the subject of left operands, as well as the source and format of their values (cf. Section 3.1);
- a unified evaluation algorithm based on comparative semantics (cf. Section 3.2); and

¹²Other examples of updates to the state of the world include the dynamic/transformational aspects of policy changes triggered by events; e.g., a person withdrawing their consent (“opting out”) [7, 1].

¹³The test cases – 68 at the time of writing – describe a state of the world, a policy, and (optionally) a request, together with an expected output – serialized using the Compliance Report Model.

- a standardization of the input and output of such an algorithm, in particular a formalized state of the world (cf. Section 3.3).

Funding Acknowledgements

This research was funded by Interreg project SecuWeb (0100085) and by the imec.icon project PACSOI (HBC.2023.0752), which was co-financed by imec and VLAIO and brings together the following partners: FAQIR Foundation, FAQIR Institute, MoveUP, Byteflies, AContrario, and Ghent University – IDLab.

Declaration on Generative AI

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

References

- [1] M. G. Kebede, G. Sileno, T. Van Engers, A Critical Reflection on ODRL, in: *Lecture Notes in Computer Science*, volume 13048 of *AI Approaches to the Complexity of Legal Systems XI-XII*, Springer, 2021, pp. 48–61. doi:10.1007/978-3-030-89811-3_4.
- [2] A. H. Anderson, A comparison of two privacy policy languages: EPAL and XACML, in: *Proceedings of the 3rd ACM Workshop on Secure Web Services*, ACM, 2006, pp. 53–60. doi:10.1145/1180367.1180378.
- [3] W3C POE WG, ODRL 2.2 Information Model, TR, 2024. URL: <https://w3.org/TR/odrl-model/>.
- [4] W3C POE WG, ODRL 2.2 Vocabulary & Expression, TR, 2024. URL: <https://w3.org/TR/odrl-vocab/>.
- [5] W. Termont, Authorization for Data Spaces, TR, Knowledge on Web Scale (IDLab), 2025. URL: <https://spec.knows.idlab.ugent.be/A4DS/L1/latest/>.
- [6] W3C ODRL CG, Issues, n.d. URL: <https://github.com/w3c/odrl/issues/>.
- [7] N. Fornara, M. Colombetti, Using semantic web technologies and production rules for reasoning on obligations, permissions, and prohibitions, *The European Journal on Artificial Intelligence* 32 (2019) 319–334. doi:10.3233/AIC-190617.
- [8] J. O. Salas, P. Pareti, S. Yumuşak, S. Gheisari, L.-D. Ibáñez, G. Konstantinidis, Evaluation and comparison semantics for ODRL, 2025. URL: <https://arxiv.org/abs/2509.05139>. arXiv: 2509.05139, (Accepted as a full paper at the 14th International Joint Conference on Knowledge Graphs (IJCKG 2025)).
- [9] W. R. van Hage, V. Malaisé, R. Segers, L. Hollink, G. Schreiber, Design and use of the Simple Event Model (SEM), *Journal of Web Semantics* 9 (2011) 128–136. doi:10.1016/j.websem.2011.03.003.
- [10] M. Rinne, E. Blomqvist, R. Keskisärkkä, E. Nuutila, Event processing in RDF, in: *Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns*, volume 1188, CEUR, 2013. URL: https://ceur-ws.org/Vol-1188/paper_1.pdf.
- [11] 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 dataspace, in: *Proceedings of the Second International Workshop on Semantics in Dataspace (SDS 2024)*, volume 3705 of *CEUR Workshop Proceedings*, CEUR, 2024. URL: <https://ceur-ws.org/Vol-3705/paper10.pdf>.
- [12] W. Slabbinck, J. A. Rojas, B. Esteves, R. Verborgh, P. Colpaert, May the FORCE be with you? A framework for ODRL rule compliance through evaluation, in: *Joint Proceedings of the 21st International Conference on Semantic Systems*, volume 4064 of *CEUR Workshop Proceedings*, CEUR, 2025. URL: <https://ceur-ws.org/Vol-4064/NXDG25-paper6.pdf>.
- [13] D. Mustafa, D. Collarana, Y. Peng, R. Haque, C. Lange-Bever, C. Quix, S. Decker, Axis Decomposition for ODRL: Resolving Dimensional Ambiguity in Policy Constraints through Interval Semantics, 2026. doi:10.48550/arXiv.2602.19878.

- [14] M. De Vos, S. Kirrane, J. Padget, K. Satoh, ODRL policy modelling and compliance checking, in: Proceedings of the Third International Joint Conference on Rules and Reasoning, volume 11784 of *Lecture Notes in Computer Science*, Springer, 2019. doi:10.1007/978-3-030-31095-0_3.
- [15] Gaia-X, ODRL profile for attribute based access/usage control using verifiable credential claims, 2024. URL: <https://gitlab.com/gaia-x/lab/policy-reasoning/odrl-vc-profile>.
- [16] A. Cimmino, N. Fornara, Improving ODRL 2.2: Current limitations and theoretical solutions, in: Joint Proceedings of the ESWC 2025 Workshops and Tutorials, volume 3977 of *CEUR Workshop Proceedings*, CEUR, 2024. URL: <https://ceur-ws.org/Vol-3977/OPAL2025-6.pdf>.
- [17] W3C ODRL CG, ODRL Formal Semantics, Draft CG Report, 2026. URL: <https://w3c.github.io/odrl/formal-semantics>.
- [18] V. Rodríguez-Doncel, N. Roman, Towards conformance in ODRL 3.0, in: Joint Proceedings of the ESWC 2025 Workshops and Tutorials, volume 3977 of *CEUR Workshop Proceedings*, CEUR, 2025. URL: <https://ceur-ws.org/Vol-3977/OPAL2025-8.pdf>.
- [19] P. A. Bonatti, N. Fornara, A. Harth, Towards a Formal Semantics of the Open Digital Rights Language (ODRL 2.2), in: Joint Proceedings of the ESWC 2025 Workshops and Tutorials, volume 3977 of *CEUR Workshop Proceedings*, CEUR, 2025. URL: <https://ceur-ws.org/Vol-3977/OPAL2025-4.pdf>.
- [20] E. I. Papagiannakopoulou, N. L. Dellas, G. V. Lioudakis, M. N. Koukovini, A. Mousas, Semantic conflict resolution for access and usage control, in: Joint Proceedings of the 21st International Conference on Semantic Systems, volume 4064 of *CEUR Workshop Proceedings*, CEUR, 2025. URL: <https://ceur-ws.org/Vol-4064/NXDG25-paper4.pdf>.
- [21] B. Esteves, W. Slabbinck, Y. Sellami, A. Cimmino, V. Rodríguez-Doncel, R. Verborgh, Capturing requests and context for ODRL-based access and usage control, in: 16th Workshop on Ontology Design and Patterns, 2025. URL: <https://ceur-ws.org/Vol-4093/paper5.pdf>.
- [22] B. Parducci, H. Lockhart, E. Rissanen, eXtensible Access Control Markup Language (XACML) Version 3.0 plus errata 01, 2013. URL: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html>.
- [23] V. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, NIST SP 800-162: Guide to Attribute Based Access Control (ABAC) Definition and Considerations, 2019. doi:10.6028/NIST.SP.800-162.
- [24] W. Slabbinck, J. A. Rojas, B. Esteves, P. Colpaert, R. Verborgh, Interoperable interpretation and evaluation of ODRL policies, in: *The Semantic Web*, Springer Nature, 2025. doi:10.1007/978-3-031-94578-6_11.
- [25] W. Slabbinck, B. Esteves, Framework for ODRL Rule Compliance through Evaluation (FORCE), TR, Knowledge on Web Scale (IDLab), 2025. URL: <https://spec.knows.idlab.ugent.be/force/latest/>.