January 2020

A critical reflection on ODRL

Milen G. KEBEDE [a,1], Giovanni SILENO [a] and Tom VAN ENGERS [b,a]
[a] *Informatics Institute, University of Amsterdam, Amsterdam, Netherlands*
[b] *Leibniz Institute, University of Amsterdam/TNO, Amsterdam, Netherlands*

Abstract. Rights expressions languages (RELs) express and govern legally binding behavior within technological environments. The Open Digital Rights Language (ODRL), used to represent statements about the usage of digital assets, is amongst the most popular RELs today and has become a W3C recommendation to enhance the web's functionality and interoperability. This paper reflects on the representational power of ODRL from a practical perspective; utilizing use cases and examples, we discuss the challenges, issues, and limitations we came across while investigating the language as a potential solution for the regulation of data-sharing infrastructures.

Keywords. Policy expression languages, ODRL, Normative specifications, Data-Sharing Infrastructures,

1. Introduction

Data usage control is one of the mechanisms that enables data owners to exercise data sovereignty (and, more generally, any party holding certain rights on data to exercise those rights). Data sharing agreements and licenses specify how, by who, for what purpose, and under which conditions data may be used or reused. In distributed data sharing infrastructures, those policies, governing, e.g., use of personal data, need to be expressed in a machine-readable knowledge representation language to support enforcement in all nodes; otherwise, they could not be applied systematically, increasing risks of non-compliance. Automating these policies fosters better transparency and the audibility of activities and inter-organizational transactions at the organizational level.

Rights expression languages (RELs) have been proposed for representing policies and utilized for specifying *digital rights* in different domains of application [1]. The primary function of those rights is to manage and protect digital assets. Several RELs have been developed, among which are the Open Digital Rights Language (ODRL), the Extensible Access Control Markup Language (XACML), the Enterprise Privacy Authorization Language (EPAL) [2]. In this paper, we focus on the Open Digital Rights Language (ODRL), which has become a *de facto* standard in the semantic web community w.r.t normative statements on data rights, reaching the status of W3C recommendation [3]. The language is presented as neutral to the technology used to grant access and is flexible enough to create new actions and constraints.

[1] Corresponding Author: m.g.kebede@uva.nl. This work have been supported by the NWO-funded projects EPI and DL4LD.

In recent years ODRL has gained popularity both in theoretical and practical settings. As our use cases focus on automating data-sharing agreements in the context of healthcare and logistics research, we found the language to be of interest and relevant to our research. Previous works have investigated the language's suitability for different scenarios and from different perspectives, and some have proposed extensions [4,5,6]. This paper shares similar motivations, although our analysis focuses on the general modeling process and requirements, as a practitioner aiming to model a policy in ODRL. Besides, we consider vital institutional patterns that were only partially covered before, as *delegation*. Delegation is a particularly relevant (and delicate) institutional construct as it brings to the foreground the requirements of meeting the needs of stakeholders while maintaining accountability.

This work aims to identify the current challenges in using ODRL for specifying policies, elaborating on the experience acquired on our use cases. The paper is structured as follows. In the next section, we provide some references to related work. Section 2 gives an overview of the core of the ODRL language, and in section 3, we report our practical investigation of the language. We conclude with a discussion in section 4.

1.1. Related work

The ODRL language has passed through several iterations, and the language maintainers and developers have shown an apparent openness to feedback from the community. The content of contributions in the literature on ODRL range from suggested extensions of the informational model, typically motivated by specific application domains, formal specifications of the language, to mappings of the language to other languages.

De Vos, Marina et al. [4] propose the application of an extended/revised ODRL model to capture the semantics of legal regulations such as the GDPR and organizational business policies. The policy profile they propose, the ”regulatory compliance profile,” can be used to model regulatory requirements and business policies via nested permissions, prohibitions, obligations, and dispensations. Shakeri et al. [5] consider the use of the ODRL in the context of digital data markets (DDMs). They extend the ODRL model by defining categories of assets and adding the input property. The first helps solve the inconvenience of defining rules for every asset in the digital data market, while the second allows defining the data used as input for data processing. Fornara et al. [6] extend the ODRL model in two directions: by inserting the notion of *activation event/action*, and by considering the temporal aspects of the deontic concepts (permission, obligation, and prohibition) as part of the application-independent model. The activation event/action notion is further expressed by events/actions as complex constructs having types and application-independent properties.

There are relatively few research efforts made towards the formalization of the semantics of the ODRL. An early work by Garcia et al. [7] formalizes the implicit semantics of ODRL schemas and connecting it to another ontology, the IPRonto. They conclude that their approach can make semantic queries possible and enable specialized reasoners over licenses. Another work by Steyskal et al. [8] addresses ambiguities that might emerge based on explicit or implicit dependencies among actions. They propose an interpretation of ODRL policy expressions' formal semantics to enable rule-based reasoning over a set of policies. The work by Hutchison et al. [9] extends ODRL and XrML, another REL that allows content authors to set access control rights to their content. The extensions enable end-users to request the modification of their current rights and for the rights-holders to grant or refuse the request. Steyskal et al. [10] demonstrate the ODRL ability to express a large variety of access policies for linked data through different examples. The work aims to mitigate issues with linked data regarding expressive access policies, introducing pricing models for online datasets, and providing a human and machine-readable form for metadata descriptions.

RELs are also used for governance in multimedia assets and intellectual property protected content. Rodriguez-Doncel et al. [11] presents the MPEG-21 contract ontology (MCO), a part of the standard ISO/IEC 21000. MCO is an ontology that represents contracts that describe rights on multimedia assets and intellectual property protected content. It describes the contract model and key elements such as the parties in the contract and the relevant clauses conveying permissions, obligations and, prohibitions. Another work by Rodriguez-Doncel et al. [12] presents a dataset of licenses for software and data, expressed as RDF for use with resources on the web. They use ODRL 2.0 to describe rights and conditions present in licenses. It provides a double representation for humans and machines alike and can enable generalized machine-to-machine commerce if generally adopted.

Some effort has been made towards modeling delegation policies (our central scenario in this paper) using the ODRL language. The work of Grunwel et al. [13] focuses on an information accountability framework that uses ODRL to model policies for delegation. In their work, they conclude that ODRL meets the requirement to model delegation policies, given that constraints and duties can be used to express the party to whom access is delegated, expiration of the access, and the types of actions. Thus far, the studies presented to support the ODRL language as either capable of handling a specific use case or extending it based on their use-case requirement. Our approach differs in that we take a broader view to identify the challenges for future extensions of the language.

2. Modeling with ODRL

The Open Digital Rights Language (ODRL) is designed as a policy expression language, aiming to provide a flexible and interoperable information model, vocabulary, and encoding mechanism for representing normative statements concerning digital content and services [3]. It has been evolving through the years from a digital rights expression language for expressing simple licensing mechanisms for the use of digital assets to accommodating privacy policies [14]. The W3C currently supports the ODRL Information Model 2.2 Recommendation. The model is built using Linked Data principles; however, all its semantics is described informally as no formal specification is provided. In the remainder of this section, we provide an overview of the ODRL information model, focusing on the main classes that are of interest for our purposes (see Fig. 1).

Overview of core ODRL classes The Open Digital Rights Language (ODRL) is designed as a policy expression language, aiming to provide a flexible and interoperable information model, vocabulary, and encoding mechanism for representing normative statements concerning An **Asset** is a digital resource that might be subject to a Rule. It has an *asset identifier* property and can be any form of identifiable resource. A **Party** refers to an entity such as a person, organization or collection of entities that undertake roles in a rule. It should have a party identifier. An **Action** class represents operations that can be

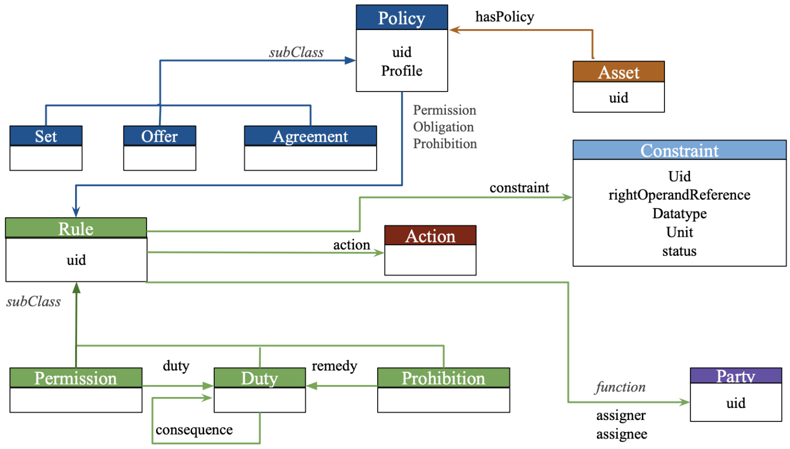

Figure 1. Simplified view on the Information Model of ODRL 2.2

exercised on assets; the association with the asset is specified via the *action* property in a rule. The **Constraint** class refines the specification of action or declares the conditions applicable to a rule by using an expression that compares two operands with an operator. When the comparison returns a match, it is considered satisfied. It has a *constraint* identifier, a *right-operand* property value data type of the right operand, a unit used in the *right operand* and the status property generated from the *left operand* action.

The **Rule** class is a superclass collecting the common characteristics of the three types of normative statements considered in ODRL: permission, prohibition, and duty. It concerns an action, which might be further refined. It must contain a *target* property (indicating the asset subjected to the rule), and might have an *assignee* and *assigner* properties (linking the rule to the associated parties). A **Permission** allows an action over an asset if all constraints are satisfied and if all duties are fulfilled. It may include one or more **duty** property values. A **Prohibition** disallows an action over an asset if all constraints are satisfied. The **remedy** property may be used when an action infringes on the prohibition. A **Duty** is the obligation to exercise an action. It is fulfilled when all constraints and refinements are satisfied and have been exercised. It may have the **consequence** property, which is an additional duty that must be fulfilled in case of violation.

A **Policy** collects a group of rules (at least one) and can be qualified as Set, Offer and Agreement. It has a unique identifier, should have at least one rule, and a *profile* property to identity the ODRL profile the policy conforms to. An ODRL profile is defined to provide a shared semantics related to a specific community need. A *set* supports expressing generic rules without further instantiating the parties involved. An *offer* supports ‘offerings’ of rules from assigner parties—it is used to make available policies to a wider audience but does not grant any rules. It specifies one party, the assigner, not the assignee. An *agreement* supports granting of rules from assigner to assignee parties and is typically used to grant the terms of the rules between the parties. Therefore, an Agreement will specify both assigner and assignee parties. content and services [3]. It has been evolving through the years from a digital rights expression language for expressing simple licensing mechanisms for the use of digital assets to accommodating privacy policies [14]. The W3C currently supports the ODRL Information Model 2.2 Recommendation.

The model is built using Linked Data principles; however, all its semantics is described informally as no formal specification is provided. In the remainder of this section, we provide an overview of the ODRL information model, focusing on the main classes that are of interest for our purposes (see Fig. 1).

3. Criticalities of ODRL

The following section will report on our experience concerning the use of ODRL in modeling patterns relevant to data-sharing agreements, highlighting the issues that emerged in the exercise. We wrote the scenarios with respect to the ODRL documentations for the information model[2], informal semantics, use-cases and vocabulary of the language[3].

3.1. Illustrative use case: delegation

There are several reasons why someone may require another person to act on their behalf in data-sharing scenarios. For example, a guardian can act on behalf of a minor; or a carer can act on behalf of a person unable to grant or deny access to data. Similar patterns occur at the level of institutes. Research institutes might grant rights to be reused by partner institutes under certain conditions to promote a shared research goal. Here, we will consider the institutional delegation scenario: *Suppose CompanyX, an institution in the Netherlands, maintaining a registry of patient data, forms a data-sharing agreement with CompanyY, an institution in Belgium. The data-sharing agreement grants permission to access the data and the possibility of delegating this permission to some other institution CompanyZ. This means that CompanyZ can be allowed to have access to the data as well.* In general cases, there might be several conditions that limit permission and delegations; these refinements will be neglected for now.

ODRL provides two main higher-level actions: `transfer` and `use`. According to the ODRL vocabulary `use` actions refers to any use of the asset (e.g. "play" music or "read" file), while the `transfer` actions explicitly refers to the transfer of ownership of the asset (lost by the agent, gained by the recipient) in its entirety (e.g. "sell" or "give"). The simplest form of delegation maps then to a transfer action as shown in listing 1:[4]

```
"@type": "agreement",
"permission":
    "assigner": "CompanyX", "assignee": "CompanyY",
    "action": "transfer", "target": "datasetA"
```

Listing 1: Delegation as transfer.

The code above is an *agreement* (that is, in ODRL terms, there is an assigner and assignee) between *CompanyX* and *CompanyY*, for transferring (ownership of) datasetA from *CompanyX* to *CompanyY*. Ownership here is assumed to include the possibility of transferring the asset again to someone else (e.g., *CompanyZ*). This model can be used as

[2] `https://www.w3.org/TR/odrl-model/`

[3] `https://www.w3.org/TR/odrl-vocab/#bib-odrl-model`

[4] The original code is in JSON and it is available at `http://grotius.uvalight.net/ODRL-policies`. For space reasons, here we will omit accolades, use indenting for nested lists, empty lines to separate policies.

a specification for *non-monotonic* delegation, where the grantor loses the permission just delegated. However, with this model, we can not specify *monotonic* delegation scenarios where the grantor maintains the delegated permission.

This especially becomes problematic to capture the power relationship between parties; e.g., the party in power has to maintain ownership of the asset, or ”veto” power to either constrain or revoke granted rights, as well as the power to transfer and/or lose ownership of the asset entirely. For these limitations, we considered the following alternative model:

```
"@type": "agreement",
"permission":
    "assigner": "CompanyX", "assignee": "CompanyY",
    "action": "grantUse", "target": "datasetA",
    "duty": [{ "action": "nextPolicy", "target": "ex:newPolicy" }]

"@type": "set",
"uid": "ex:newPolicy",
"permission": [{ "action": "read", "target": "datasetA" }]
```

Listing 2: Delegation as granting conditional usage.

In the code above, a combination of actions is used to restrict the permission to use the target asset *datasetA*. The action `grantUse` enables the assignee to create policies about the target asset (whose implicit owner is the assigner) for third parties (so it provides an implicit form of institutional power) and is recommended in the ODRL vocabulary to be used with the `nextPolicy` action. In this way, however, usage rights are restricted only to a third party and not further. In some cases, we might need to allow delegated parties to delegate. A possible model (possibly abusing the intended use of `grantUse`) would be the one expressed below:

```
"@type": "agreement",
"permission":
    "assigner": "CompanyX", "assignee": "CompanyY",
    "action": "grantUse", "target": "datasetA",
    "duty": [{ "action": "nextPolicy", "target": "ex:newGrantPolicy" }]

"@type": "set",
"uid": "ex:newGrantPolicy",
"permission":
    "action": "grantUse", "target": "datasetA",
    "duty": [{ "action": "nextPolicy", "target": "ex:newPolicy" }]

"@type": "set",
"uid": "ex:newPolicy",
"permission": [{ "action": "read", "target": "datasetA" }]
```

Listing 3: Delegation as nested granting of conditional usage.

This extension may enable us to form a hierarchical structure one step further than the previous example, but it would still not represent the full transfer of delegating power to a chain of delegators of unspecified length.

Other relevant aspects of delegation, e.g. the *revocation* of rights, also can not be specified within ODRL. Yes, expressions in ODRL provide terms for specifying deadlines or expiration dates, using the constraint class, but do not consider updating activities. To conclude, the current ODRL fits some delegation scenarios, but it lacks expressiveness to accommodate others. Additionally, these excerpts raise some concerns concerning *programmability*: the intricate forms to specify these models make it difficult to identify the standard reusable components, and obscure the fact that we are dealing with a delegation pattern.

3.2. Additional issues

In this section we address additional limitations of ODRL that we have identified during our modeling experience.

3.2.1. Ambiguous semantics for duty

Duty in its common legal sense is an action that an agent is obliged to do; otherwise, there will be a violation (see, e.g., Hohfeld's framework of primitive legal concepts [15]). In principle, the duty class provides this concept, e.g., in Listing 3, with an obligation rule:

```
"@type": "agreement",
"obligation":
   "assigner": "CompanyX", "assignee": "CompanyZ",
   "action": "compensate",
   "refinement":
      "leftoperand":"payAmount", "operator": "eq",
      "rightOperand" {"@value": "2000.00", "@type": "xsd:decimal"},
      "unit": "http://dbpedia.org/resource/Euro"
```

Listing 4: Duty class in a policy with obligation rule

The policy above states that *CompanyX* assigns to *CompanyY* the duty of compensating the former with 2000 euro. However, with a non-intuitive terminological overlap, a permission rule (i.e., a rule containing a permission property) contains an inner `duty` property (2.6.5)—linking to an instance of duty class—that in ODRL serves as a precondition for acquiring the permission:

```
"@type": "agreement",
"permission":
   "assigner": "CompanyX", "assignee": "CompanyY",
   "action": "use", "target": "datasetA",
   "duty":
      "action":"pay",
      "refinement":
         "leftoperand": "payAmount", "operator": "eq",
         "rightoperand": {"@value": "500.00", "@type": "xsd:decimal"},
         "unit": "http://dbpedia.org/resource/Euro"
```

Listing 5: Duty property in a policy with permission rule

In the policy above, *CompanyX* permits *CompanyY* to use *datasetA*, conditionally to *CompanyY* paying 500 euros. *CompanyY* has a choice. The company can choose not to pay and disregard access or pay and then acquire permission to use datasetA. Looking at

Hohfeld's theory again [15], the position of *CompanyY* would not be a duty, but rather an *institutional power*: by performing the action described in the "duty" property, the assignee will enjoy the permission. Note that, for making the policy-relevant, an implicit assumption needs to be introduced here: that the use of the data is forbidden in general. This also pinpoints another issue. If we are accepting the interpretation of this duty object as a precondition, it is not clear whether the *consequence* property (meant to trigger compensation measures to violation) can be used here: if the precondition is not satisfied, then the permission does not hold, so there cannot be a violation.

3.2.2. Lack of Granularity in identifying parties

The ODRL language considers only two functional roles for agents (assignor and assignee), but this raises several concerns. First, it is not clear if the assigner counts as the policy's creator and/or as the claim-holder (correlative of the duty-holder/assigner). Second, the roles relevant to norms and roles relevant to actions can be entirely disjoint: e.g., the party to which the duty is assigned can be different from the party that might produce the performance removing a duty. For instance, a carer might have the duty to perform a particular check-in due time. Indeed, some actions in the ODRL vocabulary have refinements that enable to specify performer and recipient roles (e.g., `trackingparty`, `"trackedparty"` for the "track" action), but these are *ad-hoc* solutions, whereas a systematic approach, e.g., based on thematic roles of action, would instead enhance readability and re-usability of patterns for different interactions.

3.2.3. Transformational aspects

The activation or revocation of rules is a critical dimension in normative reasoning. Deontic relations are not fixed and change with interactions among parties. ODRL suggests to use the constraint class where temporal and contextual information can be specified to activate or terminate rules; it also provides the consequence and remedy class for enforcing actions against violations, but this is not always sufficient. For instance, regulations such as the GDPR place great importance on data subject rights; in data sharing scenarios, patients have the right to grant, change, or revoke their consent. Changes such as those consequent to patients withdrawing their consent (i.e., triggered by action) need to be captured to maintain lawful data processing. Furthermore, change also occurs at the level of parameters of policies. Suppose CompanyX has to pay 10% of a specific fee up to the end of 2020, and some action is possible that modifies the percentage to be paid. Based on our experience with ODRL, it not possible to represent this mechanism, as it lacks a general approach to define in a machine-readable way the semantics of actions in terms of institutional or extra-institutional effects.

3.2.4. Handling conflicts

ODRL provides a strategy to resolve conflicts that arise when merging policies due to policy inheritance [16]. It uses the `conflict` property which can take either the `perm`, `prohibit` alternatively, `invalid` values to decide which rule takes precedence over the other. For example, if the conflict property is set to "perm," then the permission will override the prohibition.

While this is one way to handle conflict between rules, for more complex scenarios, other factors such as attributes of the parties and contextual information can provide a richer input for setting the conflict property. The norm in Listing 6 states that one can not share data with an institute residing outside of the EU, but if that country has a cross-border agreement with the EU and the purpose for sharing data is an emergency (e.g., an outbreak), you may share data with this institute.

```
"@type": "agreement ",
"prohibition":
    "action": "share", "target": "datasetA",
    "constraint":
        "leftOperand": "spatial", "operator": "neq",
        "rightOperand": "https://www.wikidata.org/wiki/Q458"
"permission":
    "action": "share", "target": "datasetA",
     "refinement":
          "and": { "@list": [{"@id": "ex:c1"}, {"@id": "ex:c2"}] }

"@type": "constraint", "uid": "ex:c1",
"leftOperand": "purpose", "operator": "eq",
"rightOperand": {"@value":"emergency", "@type":"xsd:string"}

"@type": "constraint", "uid": "ex:c2",
"leftOperand": "recipient", "operator": "eq",
"rightOperand": {"@value":"partOfcrossborderAgreement", "@type":"xsd:string"}
```

Listing 6: Conflict property set to `Perm` indicating permission overrides prohibition.

This example shows that taking into account contextual information in the constraints for the evaluation strategy is a more reasonable choice (rather than a static, abstract `conflict` property), as this allows to implement principles as *lex specialis*. Yet, we would need still additional mechanisms for *lex posterior* and *lex superior*.

3.2.5. Additional limitations

So far,we discussed a focused selection of the considerations we drew over our interaction with ODRL, but we acknowledged additional challenges, here reported only succinctly. Normative statements are here just about actions, but often regulations are about the outcome. For instance, a specific data processing can be licit (i.e., permitted) as performed on public sources, but the output (e.g., discriminatory decision-making) might still be illicit. Second, there are instances where action might result in creating a new asset. For example, a rule might state that *"If an asset is copied, it must be attributed to a certain party"*. The rule on the original asset needs to be modified when it is copied. These changes in activity need to be reflected in the rules. Third, the higher-level distinction between use and transfer actions is simplistic, even if considering only digital assets. Looking at transfer only in terms of ownership does not allow to consider, e.g., physical movement of data from one premise to another without changing the data rights-holder.

Finally, ODRL does not provide an exact model of the policy *life-cycle*, which has a potential application value as it would enable to capture of policy design patterns. Suppose a company makes an offer for the use of their dataset under a certain payment. If another company takes up the offer, then the policy should evolve to an agreement. It is not clear from the information model whether and how the ODRL will express these changes.

4. Discussion and conclusion

The ODRL has come a long way from its initial conception, and its success can be explained by being an accessible and powerful language. In this paper, we have addressed some significant limitations we identified on the current version of the ODRL language: the lack of monotonicity in representing delegation scenarios, semantic ambiguity in the usage of ”duty,” granularity in identifying parties, and transformational aspects of rules.

In contrast with our arguments, Grunwell et al. [13] conclude that the ODRL meets the requirements for representing access delegation policies. As stated by their work, the requirements for delegation policy are easy revocation, time dependency, and granularity. While this might be true for static policies, it did not match our experience with dynamic policies. One such example is a consent management scenario where a patient can revoke their consent at any given time. There is no mechanism to represent such events. We also found that the language is not granular enough to identify parties concerning the deontic rules. Fornara et al. [6] claim that the expressivity of their work is the same as the ODRL model in that it is possible to express deontic relations using both models. In our work, we found that the expressivity of the ODRL language in expressing deontic relations is not enough in case more roles emerge other than just an assignee and assigner or creditor and debtor [6]. They also consider the life cycle of the rules while our work covers the life cycle of a policy. Steyskal et al[10] demonstrate that ODRL is suitable to express access policies for linked data by providing different examples. One of the scenarios covered was the introduction of payment duties. They illustrate that duty assignment can be easily defined while finding the semantics of duty ambiguous and not explicit enough to express simplified assignments. Even more, we found that the semantics to show the modification of actions are missing, and this is necessary, particularly in payment scenarios (e.g., for changing rates).

In our future work, we will focus on leveraging our work on the policy language eFLINT [17], in order to mitigate some of the issues covered here. An essential aspect of eFlint is that it is action-based language and derives normative positions of actors from the actions they perform(permission) or expected to perform(duties) at a given moment, which simplifies the compliance checking of scenarios or software implementations as they are inherently action-based. Normative aspects of the language are based on the legal framework constructed by Hohfeld by integrating both core aspects of Hohfeld’s framework, i.e., describing ’normative relations’ rather than individual positions and allowing normative relations to change over time by the effects of actions and events. The normative positions of actors evolve as actions are performed, and events take place. It also supports the legal concept of power- the ability to grant or remove permissions and duties assigned.

These features can mitigate some limitations of ODRL we discussed above, such as the representation of delegation and the above mentioned transformational aspects. Because eFlint is suited to describe a wide variety of normative sources such as laws, regulations, policies, and contracts, the next step is to validate the language capability and feasibility with several use-cases from finance, healthcare, and other data marketplaces. The language also provides online execution/debugging environments and analysis features such as a query language on the running instances. We also want to perform a systematic comparison between ODRL and eFlint to extract common underlying models and test whether the interoperability of the two is feasible. Our vision with respect to an integration of ideas from ODRL and eFLINT is the development a self-contained policy specification language that is not strongly dependent on the application, or the implementation framework.

References

[1] Pramod A. Jamkhedkar, Gregory L. Heileman, and Iván Martínez-Ortiz. The problem with rights expression languages. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 59–67, 2006.

[2] Tassilo Pellegrini, Andrea Schönhofer, Sabrina Kirrane, Simon Steyskal, Anna Fensel, Oleksandra Panasiuk, Victor Mireles-Chavez, Thomas Thurner, Markus Dörfler, and Axel Polleres. A genealogy and classification of rights expression languages - Preliminary results. *Jusletter IT*, (February):1–8, 2018.

[3] Renato Iannella and Serena Villata. ODRL information model 2.2. *W3C Recommendation*, 2018.

[4] Marina De Vos, Sabrina Kirrane, Julian Padget, and Ken Satoh. Odrl policy modelling and compliance checking. In *International Joint Conference on Rules and Reasoning*, pages 36–51. Springer, 2019.

[5] Sara Shakeri, Valentina Maccatrozzo, Lourens Veen, Rena Bakhshi, Leon Gommans, Cees De Laat, and Paola Grosso. Modeling and matching digital data marketplace policies. *Proceedings - IEEE 15th International Conference on eScience, eScience 2019*, pages 570–577, 2019.

[6] Nicoletta Fornara and Marco Colombetti. Using Semantic Web technologies and production rules for reasoning on obligations, permissions, and prohibitions. *AI Communications*, 32(4):319–334, 2019.

[7] Roberto García, Rosa Gil, Isabel Gallego, and Jaime Delgado. Formalising ODRL semantics using web ontologies. *ODRL 2005 - 2nd International Open Digital Rights Language Workshop 2005*, (May 2014), 2005.

[8] Simon Steyskal and Axel Polleres. Towards formal semantics for ODRL policies. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9202:360–375, 2015.

[9] Andrew Hutchison. Extending ODRL and XrML to Enable Bi-Directional Communication. (July 2008), 2014.

[10] Simon Steyskal and Axel Polleres. Defining expressive access policies for linked data using the ODRL ontology 2.0. *ACM International Conference Proceeding Series*, 2014-Septe(September):20–23, 2014.

[11] Víctor Rodríguez-Doncel, Jaime Delgado, Silvia Llorente, Eva Rodríguez, and Laurent Boch. Overview of the mpeg-21 media contract ontology. *Semantic Web*, 7(3):311–332, 2016.

[12] Victor Rodriguez-Doncel, Serena Villata, and Asunción Gómez-Pérez. A dataset of rdf licenses. In *JURIX*, pages 187–188, 2014.

[13] Daniel Grunwel and Tony Sahama. Delegation of access in an information accountability framework for eHealth. *ACM International Conference Proceeding Series*, 01-05-Febr, 2016.

[14] Randike Gajanayake and Sahama Tony Iannella Renato. An information accountability framework for shared ehealth policies. 2012.

[15] Wesley Newcomb Hohfeld. Fundamental legal conceptions as applied in judicial reasoning. *The Yale Law Journal*, 26(8):710–770, 1917.

[16] Erisa Karafili and Emil C. Lupu. Enabling Data Sharing in Contextual Environments. pages 231–238, 2017.

[17] L. Thomas van Binsbergen, Robert van Doesburg, and Tom van Engers. eFLINT : a Domain-Specific Language for Executable Norm Specifications. *Proceedings of GPCE '20. ACM.*