A Critical Reflection on ODRL

Kebede, M.G.; Sileno, G.; Van Engers, T.

DOI
10.1007/978-3-030-89811-3_4

Publication date
2021

Document Version
Final published version

Published in
AI Approaches to the Complexity of Legal Systems XI-XII

A Critical Reflection on ODRL

Milen G. Kebede[1(✉)], Giovanni Sileno[1], and Tom Van Engers[2]

[1] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
{m.g.kebede,g.sileno}@uva.nl
[2] Leibniz Institute, University of Amsterdam/TNO, Amsterdam, The Netherlands
vanengers@uva.nl

Abstract. Rights expression languages (RELs) aim to 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 among the most known 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 specification · Data sharing infrastructure

1 Introduction

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

Rights expression languages (RELs) are originally proposed for representing policies and utilized for specifying *digital rights* in different domains of application [13]. The primary function of those rights is to manage and protect digital

This work is supported by the project Enabling Personalized Interventions (EPI, grant 628.011.028), funded by NWO in the Commit2Data—Data2Person program, and by the project Data Logistics for Logistics Data (DL4LD, grant 628.009.001), funded by NWO and TKI Dinalog in the Commit2Data initiative.

V. Rodríguez-Doncel et al. (Eds.): AICOL-XI 2018/AICOL-XII 2020/XAILA 2020, LNAI 13048, pp. 48–61, 2021.
https://doi.org/10.1007/978-3-030-89811-3_4

assets.[1] Several RELs exist, among which the Open Digital Rights Language (ODRL) [12], the Extensible Access Control Markup Language (XACML) [1], and the Enterprise Privacy Authorization Language (EPAL) [16].

In this paper, we focus on the Open Digital Rights Language (ODRL), a language that in recent years has gained popularity both in theoretical and practical settings, reaching the status of W3C recommendation [12]. The language is presented as being neutral to the technology used to implement usage control and is intended to be flexible enough to allow for the creation of new actions and constraints for data access policies. As our use cases focus on automating data-sharing agreements in the context of healthcare and logistics research, we found the language relevant to our research. Our goal is to utilize a language that supports the specification of normative constructs as those specified in regulations, agreements and consents. While extended versions of XACML support partial specification and enforcement of laws and regulations, it lacks, for instance, the support for "system obligations" [15], i.e. obligations the system has to perform on certain events such as notification of data breach. On the other hand, EPAL [4] is designed for writing enterprise privacy policies but it lacks e.g. reasoning support for conflicts or other relevant constructs [2].

Previous work investigated ODRL's suitability for different scenarios and from different perspectives, and also proposed various extensions [6,7,19]. The present contribution shares similar motivations, although our analysis focuses on the general modeling process and requirements taking the standpoint of a designer aiming to model a policy in ODRL. Additionally, we consider crucial institutional patterns that were only partially covered before, such as *delegation*. As an institutional construct, delegation is particularly relevant (and delicate), as it brings to the foreground the requirements of meeting the needs of stakeholders while maintaining accountability. The general aim of this paper is then to present the current challenges in using ODRL for specifying policies, elaborating on the experiences acquired on a data-sharing use case in the healthcare domain. Several limitations of the ODRL language are discussed, such as the lack of monotonicity in representing delegation scenarios, semantic ambiguity in the usage of "duty", granularity in identifying parties and transformational aspects of rules.

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 model of the ODRL language, and in Sect. 3, we report our practical investigation of the language. We conclude with a discussion in Sect. 4.

1.1 Related Work

The ODRL language passed through several iterations, and the language maintainers and developers have shown openness to feedback from the community.

[1] In the past, rights expression languages and related technologies have been criticized for resulting in stronger restrictions than what generally granted by law. However, if RELs enable to create policy layers to integrate policies derived from legislation such as e.g. GDPR, they also make it possible to counterbalance excessive protections required by businesses.

Contributions in the literature range from suggesting extensions of the ODRL informational model [12], typically motivated by specific application domains, to introducing formal specifications, and to mapping of the language to other languages.

De Vos et al. [6] 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 proposed policy profile, the "regulatory compliance profile", can be used to model regulatory requirements and business policies via nested permissions, prohibitions, obligations, and dispensations. Shakeri et al. [19] 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 to solve the inconvenience of defining rules for every asset in the digital data market, while the second allows for defining the data used as input for data processing. Fornara et al. [7] 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 ODRL. Garcia et al. [9] have formalized the implicit semantics of ODRL schemas and connected ODRL to the IPRonto ontology. They conclude that their approach can make semantic queries possible and enable specialized reasoners over licenses. Steyskal et al. [21] address 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. Arnab et al. [3] extended ODRL and XrML, a REL that allows content authors to set access control rights to their content. The extensions enable end-users to request the modification of current rights and allow rights-holders to grant or refuse the request. Steyskal et al. [20] demonstrate the ODRL's ability to express a large variety of access policies for linked data through different examples. These authors aim 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. [17] present 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. [18] 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.

There exist a few contributions aiming to model delegation policies (a central scenario in this paper) using the ODRL language. For instance, Grunwel et al. [10] focus 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.

At higher-level, the studies presented above approach ODRL taking into account a specific use case, in many cases extending the language based on the use-case requirements. Our approach differs in that we take into account a wider range of institutional constructs including duties, power, delegation and other relevant normative concepts 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 [12]. It evolved through the years from a digital rights expression language for expressing simple licensing mechanisms for the use of digital assets to accommodating privacy policies [8]. The model is built using Linked Data principles; however, 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 the institutional constructs under our attention (see Fig. 1).

Overview of Core ODRL Classes. 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 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 super-class 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

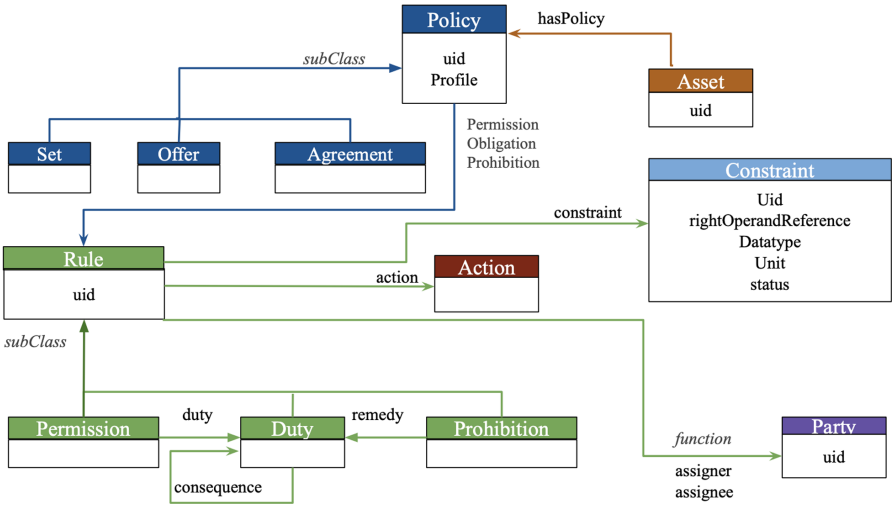

Fig. 1. Simplified view on the Information Model of ODRL 2.21

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 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 identify the ODRL profile the policy conforms to. An ODRL profile is defined to provide vocabulary terms that can be used in ODRL policies that require them, typically to be shared within a community of practice. 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. In the remainder of this section, we provide an overview of the ODRL information model, focusing on the main classes that are of interest to the use-case (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 examples with respect to the ODRL documentations on the information model[2], informal semantics, use-cases and vocabulary of the language[3].

3.1 Illustrative Use Case: Delegation

Data-sharing scenarios, at times, require individuals to act on behalf of another. For example, a guardian may be required to act on behalf of a minor; or a carer on behalf of a person unable to grant or deny access to data. Similar patterns occur at the level of institutes. Additionally, research institutes might grant rights to be used by partner institutes under certain conditions to promote a shared research goal. This section will focus on institutional delegation scenarios. For instance, *suppose OrganizationX, an institution in the Netherlands maintaining a registry of patient data, forms a data-sharing agreement with OrganizationY, an institution in Belgium. The data-sharing agreement grants OrganizationY the permission to access the data and the possibility of delegating this permission to a third party, OrganizationZ. Consequently, the latter will be allowed to have access to the data if OrganizationY decides to delegate the permission received from OrganizationX.* Several contextual information might limit permissions and delegations, typically by means of constraints; 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"). This form of delegation (in the sense of transfer of rights) maps to a transfer action as shown in listing 1:[4]

```
"@type": "agreement",
"permission":
    "assigner": "OrganizationX", "assignee": "OrganizationY",
    "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 *OrganizationX* and *OrganizationY*, for transferring

[2] https://www.w3.org/TR/odrl-model/.
[3] https://www.w3.org/TR/odrl-vocab/.
[4] The original JSON code is at https://grotius.uvalight.net/ODRL-policies. For space reasons, here we will omit accolades, use indenting for nested lists, empty lines to separate policies.

(ownership of) datasetA from *OrganizationX* to *OrganizationY*. Here, ownership is assumed to include the possibility of transferring the asset again to someone else (e.g., *OrganizationZ*). This model can be used as a specification for *non-monotonic* delegation, where the grantor loses the permission delegated. However, the same model can not be used to 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 consider the following alternative model:

```
"@type": "agreement",
"permission":
    "assigner": "OrganizationX", "assignee": "OrganizationY",
    "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 but limited form of institutional power) and is recommended in the ODRL vocabulary to be used with the `nextPolicy` action. The function of `nextPolicy` (which, to reiterate, is an action, not a policy) is to indicate the policy that applies to a third party for their use of the Asset (see e.g. [20]). In this way, however, usage rights are restricted only to a third party and not further. In some cases, delegated parties need to be allowed to delegate. A possible model (possibly abusing the intended use of `grantUse`) would be the one expressed below:

```
"@type": "agreement",
"permission":
    "assigner": "OrganizationX", "assignee": "OrganizationY",
    "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, yet it can 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. While expressions in ODRL provide terms for specifying deadlines or expiration dates using the constraint class, updating activities are not considered. To conclude, the current ODRL model fits some delegation scenarios, but lacks expressiveness to accommodate others. Additionally, 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.

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 [11]). In principle, the duty class provides this concept, e.g., in Listing 3, with an obligation rule:

```
"@type": "agreement",
"obligation":
    "assigner": "OrganizationX", "assignee": "OrganizationZ",
    "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 *OrganizationX* assigns to *OrganizationY* 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 of the ODRL Information Model)—linking to an instance of duty class—that in ODRL serves as a pre-condition for acquiring the permission:

```
"@type": "agreement",
"permission":
    "assigner": "OrganizationX", "assignee": "OrganizationY",
    "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, *OrganizationX* permits *OrganizationY* to use *datasetA*, conditionally to *OrganizationY* paying 500 euros. *OrganizationY* has a choice. The organization can choose not to pay and disregard access or pay and then acquire permission to use datasetA. Looking at Hohfeld's theory again [11], the position of *OrganizationY* is not 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. We have found no specific constraint in the ODRL Information Model.

Lack of Granularity in Identifying Parties. The ODRL language considers only two functional roles for agents (assignor and assignee), a choice which 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 produces 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 allow refinements that enable specifying 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, instead enhances readability and re-usability of patterns for different interactions.

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, OrganizationX 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 is 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.

Handling Conflicts. ODRL provides a strategy to resolve conflicts that arise when merging policies due to policy inheritance [14]. 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 data can not be shared outside of the EU, but if the recipient has a cross-border agreement and the purpose for sharing data is an emergency (e.g., an outbreak), then data may be shared.

```
"@type": "agreement",
"conflict": "perm",
"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 that permission overrides prohibition.

The example above demonstrates that conflict resolution specifications are independent from the contextual information (i.e. constraints). Rather than a static, abstract `conflict` property, a more reasonable choice would be to take constraints into consideration and to implement principles as *lex specialis*. (Additional mechanisms are also required in principle for *lex posterior* and *lex superior*.)

Additional Limitations. So far, we discussed a focused selection of the considerations we drew over our interaction with ODRL, and acknowledged additional challenges, here reported only succinctly. Normative statements are about actions, while regulations, often, are about outcomes. For instance, a specific data processing can be licit (i.e., permitted) as performed on public sources, yet 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 us 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 enables capturing 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 came a long way from its initial conception and its wide adoption can be attributed to its accessibility and expressiveness. In this paper, we addressed some significant limitations 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.

Our findings are in contrast with those of Grunwell et al. [10] that 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 hold for static policies, we have shown that this does not hold for dynamic policies as for instance the scenario in which a patient can revoke their consent at any given time. There is no mechanism to represent such events in ODRL. Fornara et al. [7] 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 representing deontic relations is not enough in cases where more roles emerge, thus exceeding the limitation of two complementary roles such as assignee and assigner or creditor and debtor [7]. Additionally, these authors consider the lifecycle of the rules (i.e. their dynamics) while our work covers the lifecycle of a policy. Steyskal et al.[20] demonstrate that ODRL is suitable to express access policies for linked data by providing different scenarios. One of the examples covered is the introduction of payment duties. They illustrate that duty assignment can be easily defined but they also find the semantics of duty ambiguous and not explicit enough to express simple assignments. We confirm their analysis, and we additionally found that the semantics to specify the modification of actions is missing, and this is necessary, particularly in payment scenarios (e.g., for changing rates).

In next steps, we will focus on studying whether the policy specification language eFLINT [5] can overcome some of the issues covered here. An essential aspect of eFLINT is that it is an action-based language and derives normative positions of actors from the actions they perform (permission) or expected to perform (duties) at a given moment. This simplifies to perform e.g. compliance checking of scenarios or software implementations as they are inherently action-based. Furthermore, normative aspects of the language are based on the framework of normative concepts proposed by Hohfeld, supporting the use of a primitive for legal power—the ability to grant or remove permissions and duties assigned. These features should mitigate some of the limitations of ODRL discussed above, such as the representation of delegation and transformational aspects. The next step is to validate the expressiveness and tractability of the language with several use-cases from finance, healthcare, and other data marketplaces.

As future work, we plan 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 of a self-contained policy specification language that is as much as possible independent on the application, or the implementation framework.

References

1. Anderson, A., et al.: Extensible access control markup language (XACML) version 1.0. OASIS (2003)
2. Anderson, A.H.: A comparison of two privacy policy languages: EPAL and XACML. In: Proceedings of the 3rd ACM workshop on Secure web services, pp. 53–60 (2006)
3. Arnab, A., Hutchison, A.: Extending ODRL and XrML to enable bi-directional communication. University of Cape Town, Technical report (2004)
4. Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise privacy authorization language (EPAL). IBM Res. **30**, 31 (2003)
5. van Binsbergen, L.T., Liu, L.C., van Doesburg, R., van Engers, T.: eFLINT: a domain-specific language for executable norm specifications. In: Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, pp. 124–136 (2020)
6. De Vos, M., Kirrane, S., Padget, J., Satoh, K.: ODRL policy modelling and compliance checking. In: Fodor, P., Montali, M., Calvanese, D., Roman, D. (eds.) RuleML+RR 2019. LNCS, vol. 11784, pp. 36–51. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31095-0_3
7. Fornara, N., Colombetti, M.: Using semantic web technologies and production rules for reasoning on obligations, permissions, and prohibitions. AI Commun. **32**(4), 319–334 (2019)
8. Gajanayake, R., Iannella, R., Sahama, T.: An information accountability framework for shared ehealth policies. In: Data Usage Management on the Web: Proceedings of the WWW2012 Workshop, pp. 38–45. Technische Universitat Munchen-Institut fur Informatik (2012)
9. García, R., Gil, R., Gallego, I., Delgado, J.: Formalising ODRL semantics using web ontologies. In: Proceedings of the 2nd International Workshop on ODRL, pp. 1–10 (2005)
10. Grunwel, D., Sahama, T.: Delegation of access in an information accountability framework for eHealth. In: ACM International Conference Proceeding Series (2016)
11. Hohfeld, W.N.: Fundamental legal conceptions as applied in judicial reasoning. Yale Law J. **26**(8), 710–770 (1917)
12. Iannella, R., Villata, S.: ODRL Information Model 2.2. W3C Recommendation (2018)
13. Jamkhedkar, P.A., Heileman, G.L., Martínez-Ortiz, I.: The problem with rights expression languages. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 59–67 (2006)
14. Karafili, E., Lupu, E.C.: Enabling data sharing in contextual environments: policy representation and analysis. In: Proceedings of the 22Nd ACM on Symposium on Access Control Models and Technologies, pp. 231–238 (2017)
15. Leicht, J., Heisel, M.: A survey on privacy policy languages: Expressiveness concerning data protection regulations. In: 2019 12th CMI Conference on Cybersecurity and Privacy (CMI), pp. 1–6. IEEE (2019)
16. Pellegrini, T., et al.: A genealogy and classification of rights expression languages-preliminary results. In: Data Protection/LegalTech-Proceedings of the 21st International Legal Informatics Symposium IRIS, pp. 243–250 (2018)
17. Rodríguez-Doncel, V., Delgado, J., Llorente, S., Rodríguez, E., Boch, L.: Overview of the MPEG-21 media contract ontology. Semant. Web **7**(3), 311–332 (2016)

18. Rodriguez-Doncel, V., Villata, S., Gómez-Pérez, A.: A dataset of rdf licenses. In: JURIX, pp. 187–188 (2014)
19. Shakeri, S., et al.: Modeling and matching digital data marketplace policies. In: Proceedings of the IEEE 15th International Conference on eScience, eScience 2019, pp. 570–577 (2019)
20. Steyskal, S., Polleres, A.: Defining expressive access policies for linked data using the ODRL ontology 2.0. In: Proceedings of the 10th International Conference on Semantic Systems, pp. 20–23 (2014)
21. Steyskal, S., Polleres, A.: Towards formal semantics for ODRL policies. In: Bassiliades, N., Gottlob, G., Sadri, F., Paschke, A., Roman, D. (eds.) RuleML 2015. LNCS, vol. 9202, pp. 360–375. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21542-6_23