



CORAL-AC Dataset: COrpus of RLS And ODRL policies for Linked Access Control

Carlos Rubio-Medina¹, Andrea Cimmino¹, Nicoletta Fornara², and Joaquin Gayoso-Cabada¹

¹ Universidad Politécnica de Madrid

`carlos.rubiom@alumnos.upm.es`, `andreajesus.cimmino@upm.es`, `j.gayoso@upm.es`

² Università della Svizzera italiana

`nicoletta.fornara@usi.ch`

Abstract

The availability of datasets containing machine-readable policies is essential for evaluating policy modelling approaches and policy decision engines. However, publicly available datasets that provide policies expressed using the Open Digital Rights Language (ODRL) remain scarce, particularly in domains that require complex contextual information and operational constraints. This paper presents the CORAL-AC dataset, a curated collection of access control policies derived from a hospital information system scenario. The dataset includes fifteen policies represented at four complementary levels: natural language descriptions, SQL Row-Level Security (RLS) policies, and semantically equivalent policies expressed using ODRL and ODRE (an ODRL-based language for policy enforcement). In addition, the dataset provides structured evaluation artefacts including states of the world, evaluation requests, and expected outcomes aligned with the W3C ODRL formal semantics work. The construction of the dataset follows a systematic methodology that starts from operational requirements and progressively externalises them into machine-readable policy representations. During this process, several challenges were identified when representing operational access control rules using ODRL, particularly regarding role resolution, filtered assets, dynamic contextual information, and temporal constraints. The paper discusses these challenges and analyses how they can be addressed using extensions based on the ODRE approach. The resulting dataset aims to support research in different areas, such as, policy modelling, recommendation, semantic policy translation, and policy engine evaluation, providing a reusable resource for the ODRL community.

1 Introduction

Healthcare information systems manage highly sensitive data, including clinical histories, medication plans, and administrative records [37]. Access to such data must be regulated according to strict organisational, ethical, and legal requirements. In practice, fine-grained access control in hospital environments is frequently enforced at the database layer using mechanisms such as Row-Level Security (RLS) [36, 31], which restricts data access through predicates evaluated directly by relational engines such as PostgreSQL and SQL Server [11]. RLS enables

administrators to define policies, i.e., tuple-level access predicates expressed as queries, that are transparently enforced during query execution.

RLS provides enforcement guarantees by preventing unauthorised queries from returning tuples that do not satisfy predefined predicates. This approach embodies a regimentation-based control mechanism, in which violations are technically prevented rather than detected after execution [3]. Although effective from an enforcement perspective in access control scenarios, RLS policies remain tightly coupled to a specific database implementation, since they are expressed procedurally as SQL predicates. Furthermore, they are not suitable for other scenarios rather than access control, e.g., usage control [2], which requires continuous monitoring of digital asset usage in decentralised environments. As a result, the semantics of the RLS policies are embedded in the execution environment, and the applications that use them are not interoperable.

The lack of semantic abstraction in native database policies raises several research challenges. First, RLS rules are not portable across heterogeneous infrastructures, which limits their applicability in federated or cross-organisational settings. Second, the intended meaning of the access constraints is not explicitly modelled but implicitly encoded in the SQL conditions, making independent analysis and formal reasoning difficult. Third, the RLS mechanisms do not expose a structured evaluation model that makes explicit the relationship between a policy, a contextual state (also known as state of the world [20]), and an access request (also known as evaluation request [20]). As a result, policies cannot be analysed or benchmarked independently from the database instance in which they are deployed.

These limitations are particularly relevant in emerging contexts such as European Data Spaces [25] and distributed healthcare ecosystems [10], where policy interoperability, transparency and auditability are required [24]. In such scenarios, access control rules must be expressed in a declarative form that enables semantic interpretation and independent evaluation.

The Open Digital Rights Language (ODRL) is a W3C recommendation designed to express permissions, prohibitions, and duties over digital resources [23, 22]. ODRL has been adopted and recommended in various domains, including Internet of Things infrastructures, Data Spaces, and Knowledge Graph governance [33, 15]. Despite its growing adoption, open research challenges are still open, such as policy enforcement [7] and descriptive and behavioural issues [6]; among others. In particular, while ODRL provides an ontology for policy representation, its practical adoption in real-world scenarios remains limited, and curated datasets grounded in operational systems, which are a vital contribution to advance in many research challenges, are scarce [9, 21].

In this paper, a curated open dataset of policies is presented as a resource, which is built on top of a synthetic use case based on hospital access control policies originally implemented using Row-Level Security. The dataset, named CORAL-AC (COrpus of RLS And ODRL policies for Linked Access Control)¹ comprises 15 policies covering different use cases: role-based access, ownership constraints, departmental restrictions, emergency access, anonymisation requirements, financial conditions, and temporal limitations. Despite emerging proposals that use LLMs to generate policies [27, 38], in CORAL-AC each policy was manually developed by experts and expressed in natural language, as an RLS query, and as an ODRL policy to ensure the quality of the policies delivered.

The dataset published in this work has two research values. First, it provides a curated set of ODRL policies accompanied by explicitly defined states of the world, evaluation requests, and

¹The CORAL-AC dataset is available at this URL: <https://github.com/ODRE-Framework/coral-ac-dataset>

expected decision outcomes. By making the evaluation structure explicit following the W3C ODRL formal semantics methodology [20], the dataset constitutes a reproducible benchmark for enforcement engines and compliance verification frameworks. Researchers can systematically assess the correctness, completeness, and behavioural alignment of ODRL evaluators without depending on the original database infrastructure. Second, each policy is explicitly linked to its equivalent Row-Level Security (RLS) rule, ODRL representation, ODRE representation, and to a controlled natural language description of its intended meaning. ODRE is a proposal that combines the ODRL vocabulary with meta-languages to introduce the concept of template and policy variables and perform policy enforcement [7]; the rationale behind this decision was to address some limitations of the ODRL policies in certain scenarios. This alignment offers a structured resource for future research on policy translation, reverse engineering, and recommendation [34, 6]. Beyond enforcement benchmarking, the dataset supports investigations into cross-language policy transformation and explainability in heterogeneous policy formalisms.

2 Related Work

Datasets that provide machine-readable policies to address different research challenges are limited. In particular, datasets containing policies expressed using ODRL are especially scarce. Focusing on the well-established XACML standard for policy specification [26], XACBench provides tools to derive statistical profiles from input policies and to generate synthetic policy sets for benchmarking evaluation algorithms. This initiative is explicitly motivated by the limited availability of real-world policy corpora [1]. In the domain of cloud access control, Quacky introduces a benchmark of AWS policies derived from real forum examples and expanded through systematic mutations, allowing quantitative analyses of permissiveness [13, 12]. More recently, benchmarking efforts have emerged in large language model centric environments, focusing on policy adherence and security constraints, where explicit policies and queries are used to evaluate compliance behaviour [35].

Focusing on ODRL-related datasets, particularly in decentralised personal data infrastructures such as Solid, ODRL has been adopted through the ODRL Profile for Access Control (OAC), which combines ODRL with Data Privacy Vocabulary (DPV²) to represent access control and consent-oriented constraints over personal data resources [17]. Building upon this approach, several contributions provide representative policy examples and demonstrate request-policy matching in concrete scenarios [16, 18]. However, these works are primarily positioned as architectural or modelling proposals rather than as curated datasets including systematic evaluation cases. Similarly, proposals related to policy enforcement [7, 32] present policies to illustrate enforcement algorithms, but they do not provide key features of the dataset introduced in this article, such as the alignment between multiple representations including natural language, RLS, and ODRL, or the inclusion of a structured formal evaluation analysis.

The dataset published by Rodríguez-Doncel et al., consisting of licenses expressed in ODRL [30], represents one of the closest related contributions. Nevertheless, this work addresses a licensing scenario rather than access control, and the dataset does not provide a formal evaluation analysis of the policies, which is essential for enforcement-oriented research. Although the licenses include a natural language description inherently linked to the ODRL representation, the focus remains on licensing semantics rather than operational access control. A related line of research concerns the automatic extraction of ODRL policies from natural language licenses, such as Creative Commons licenses, as proposed by Revel et al. [29]. Their

²<https://w3c.github.io/dpv/2.3/dpv/>

contribution aims at automatically deriving machine-readable policies from textual license descriptions, rather than publishing a curated dataset as a reusable research artefact.

It is also relevant to mention recent proposals under development [27, 38] that aim at generating synthetic ODRL policies using large language models. While these approaches are promising, they raise important questions regarding the semantic validity and correctness of automatically generated policies. In contrast, the dataset presented in this article has been curated by domain experts. Although the number of policies is smaller, emphasis is placed on semantic accuracy and structural consistency. Furthermore, these proposals focus on policy generation rather than on the publication of a benchmark-oriented dataset, even though they could potentially serve as a means to construct one.

In health and biomedical data governance, the Data Use Ontology (DUO) has become a prominent vocabulary for expressing reuse constraints [28]. Recent work explores the expression of DUO concepts as ODRL policies, including their alignment with DPV and perspectives on policy and request matching. The accompanying DUO-ODRL-DPV artefacts provide a reusable collection of ODRL policy fragments for DUO concepts and a demonstrator for Offer and Request matching [14]. These resources move closer to dataset-oriented dissemination by publishing machine-readable policy fragments and reusable patterns. However, they do not provide a systematic set of contextual states and evaluation requests with expected outcomes in a benchmark-ready form suitable for comparative evaluation across different enforcement engines. In addition, while they provide policy artefacts, often combining DUO and ODRL elements, they do not systematically include textual descriptions or a formal semantic evaluation analysis comparable to the one presented in this work.

Finally, the W3C ODRL Formal Semantics [20] is currently available as a draft that specifies the expected results provided by an *ODRL Evaluator*, a piece of software that evaluates an ODRL policies based on an *evaluation request* and taking into account a given *state of the world*; providing results regarding whether an action is permitted or whether it violates or satisfies a prohibition or an obligation. The evaluation request and the state of the world can be represented using an ontology [19]. This draft includes a formal analysis of satisfaction for a set of 6 policies. Compared to the dataset introduced in this article, the policies in the W3C document do not include an RLS representation. Moreover, the dataset presented here introduces additional scenarios, including role-based enforcement patterns, that are not yet covered in the W3C document. The formal semantics of ODRL 2.2 is also discussed in [4] where crucial remarks are made regarding the expressiveness of the ODRL policy language.

To the best of the authors' knowledge, the dataset presented in this article, which links ODRL policies with natural language descriptions, RLS representations, contextual states, and structured evaluation requests, constitutes the first resource of its kind designed to support multiple open research challenges in policy representation, enforcement, and benchmarking.

3 Hospital Policy Dataset

This section describes the methodology followed to build the dataset, high-level characteristics of the synthetic hospital scenario, and the structure and content of the policy corpus. For the sake of readability, the following RDF prefixes are defined:

3.1 Methodology

The dataset introduced in this work is based on a synthetic hospital use case designed to emulate realistic access control requirements in an eHealth information system. Although the scenario

Prefix URI

sch: <https://schema.org/>
ex: <https://hospital-knowledge-graph.org/>
db: <https://hospital-database.org/db/table/>
odrl: <http://www.w3.org/ns/odrl/2/>
coral: <https://raw.githubusercontent.com/ODRE-Framework/hospital-rls-dataset/refs/heads/main/coral.owl>

is synthetic, the policy definitions were elaborated through iterative discussions with healthcare professionals and technical staff familiar with hospital information systems. In particular, medical practitioners and system administrators were consulted to validate the plausibility of role definitions, resource boundaries, and operational constraints. The policies were elaborated by experts and reviewed by different experts in order to increase their quality. The construction process followed the following stages:

1. **Requirement elicitation.** A set of representative access control interactions was identified, covering clinical, administrative, emergency, research, and external collaboration contexts.
2. **Policy definitions.** each interaction was translated into a policy expressed in natural language.
3. **Operational grounding.** Each policy in natural language was translated into a Row-Level Security (RLS) rule in PostgreSQL to reflect how such constraints would typically be implemented in practice. These rules were manually written by RLS experts.
4. **Semantic externalisation.** Each policy expressed in RLS and in natural language was manually expressed in ODRL using JSON-LD 1.1 and ODRE syntax, preserving their intended semantics: modelling roles as instances and aligning domain entities with standard vocabularies like Schema.org³.
5. **Evaluation analysis.** For each policy, a set of *states of the world* and *evaluation requests* was defined, together with the expected theoretical evaluation results. The analysis follows the conceptual structure proposed in the W3C ODRL Formal Semantics work [20], distinguishing policy satisfaction, constraint evaluation, activation state, and control decision.
6. **Review of policies.** Policies and related artefacts were reviewed by experts different from those involved in stage 2.
7. **Dataset publication.** Finally, each of the elements elaborated in this work was packed into a dataset and published as a release on GitHub. There are two main contributions, one dataset named *policy-alignments.csv* that presents each of the 15 policies aligned in their natural language, RDL, ODRE, and ODRL representations, the other is the dataset named *evaluation-analysis.csv* that presents for each ODRL policy an evaluation analysis following the W3C ODRL Formal Semantics document [20].

The resulting dataset constitutes a curated artefact linking implementation, formal policy expression, and evaluation behaviour. The dataset can be used for several tasks: A) Policy

³<https://schema.org/>

translation between natural language and ODRL (or vice-versa) and between ODRL and SQL; B) Policy recommendation, from a set of requirements given in natural language, for instance, find relevant related policies; C) Policy evaluation, given a policy expressed in JSON-LD 1.1 an evaluation request, and, if need, a state of the world, compute which restrictions are satisfied.

3.2 Scenario and Domain Model

The use case models a medium-sized hospital information system composed of multiple functional domains. The main entities of the scenario include:

Roles (Assignees or Assigners). The dataset defines fifteen policies involving different role groups. Roles are modelled in ODRL as instances `odrl:Party` or `odrl:PartyCollection` and aligned, when possible, with the *Schema* ontology to improve semantic interoperability since the ODRL classes `odrl:Party` and `odrl:PartyCollection` (which is a subclass of the former) are subclasses of the union of several classes, among which `schema:Organization` and `schema:Person`⁴. In those cases where *Schema.org* ontology lacks a suitable class, authors extended the ontology for the specific concepts and published it along the dataset. The ontology extension is published under the prefix *coral:*. The roles defined for the assignees are explained in the repository due to space limitations.

Note that all policies have as assigner the hospital, identified by the URI `ex:hospital-organization`, which is the organisation that holds the ownership of the policies. The authors will delve into how this organisation was modelled in Section 4. The roles were defined in the CORAL vocabulary in order to represent the different actors interacting with the hospital information system.

Resources (Targets). The principal resources referred by the policies, i.e., their target, are the different tables of an SQL database, which are described in the GitHub repository due to space limitations. Since no normative standard defines a canonical URI pattern for SQL schema tables (or elements), in this article the authors follow a strategy based on well-known and good Semantic Web principles. Tables are identified through a structured namespace reflecting the database context (e.g., `https://hospital-database.org/db/table/ClinicalRecords` or `db:ClinicalRecords`), while row-level resources extend the table identifier with the corresponding primary key (e.g., `db:ClinicalRecords/12345`). Similarly, restrictions can be declared on the rows indicating them as parameters in the URL, for instance, anyone with '12345' as patient id `db:ClinicalRecords?patient_id='12345'`; or the specific columns to be returned if no value is assigned, e.g., `db:ClinicalRecords?patient_id`. This approach treats relational artefacts as first-class RDF resources that can be referenced by ODRL policies as targets and described within states of the world, while remaining independent of any specific database engine or deployment configuration.

3.3 Dataset Structure

The whole dataset is split into the following csv files and a tree directory:

1. **policy-alignments.csv** containing the alignment between the different representations of each policy, it has the following columns:

- (a) **PolicyDesc**, which briefly explains the purpose of the policy.

⁴<https://www.w3.org/ns/odrl/2/ODRL22.ttl>

- (b) **GlobalID**, the unique identifier of the policy that is used in all formal formats.
 - (c) **Policy NL**, the policy expressed in natural language.
 - (d) **Policy RLS**, the policy expressed using PostgreSQL Row-Level Security.
 - (e) **Policy ODRL**, the policy expressed using the W3C ODRL 2.2 Vocabulary.
 - (f) **Policy ODRE**, the policy expressed using the ODRE vocabulary [7].
2. **evaluation-analysis.csv** containing for each policy a set of states of the world and evaluation requests to cover all possible cases that an evaluator engine might face, it has the following columns:
- (a) **PolicyDesc**, **Global ID**, and **Policy ODRL** as in the previous csv file.
 - (b) **EvaluationRequest**, the evaluation request that describes the action to be evaluated, along with the actor performing it, the object on which it would be performed, and other relevant parameters that must be evaluated.
 - (c) **StateOfTheWorld** the contextual state of the world necessary for evaluating certain policies.
 - (d) **Outcome** the expected outcome of the evaluation.

3.4 Alignment with Formal Semantics Structure

The evaluation design follows the conceptual separation proposed in the ODRL formal semantics draft [20], where: A) Policies define normative conditions; B) The state of the world provides contextual truth assignments; C) The evaluation request instantiates concrete action that has to be evaluated; D) The evaluation produces a decision outcome. Unlike purely theoretical treatments, the dataset operationalises this structure by providing concrete, finite instances of states and requests. This makes it possible to empirically validate ODRL evaluators against expected outcomes without requiring access to the original relational database environment.

4 Policies and challenges

For reasons of space, the complete representations of the fifteen policies, as well as the full evaluation tables, are not provided explicitly in this section. Also, before delving into the challenges detected expressing the ODRL policies for this scenario, it is worth to mention that the scenario is built on the hypothesis that the hospital already has a database with the information (as aforementioned) and such data cannot be replicated outside the hospital due to privacy reasons. This restriction hinders the expression of constraints, specific identifiers, and data within their database as part of the state of the world knowledge graph for the policy.

The following subsections outline the operational challenges encountered when formalising the policies using ODRL 2.2. Some of these challenges are aligned with certain limitations of ODRL 2.2 that have been introduced in the state of the art by other research articles [8, 7]. Some of the resulting policies are not exactly what is needed for the use case since they do not fully express what the use case requires. For these policies, the authors decided to follow the ODRE approach [7] introducing new concepts, such as variables, in the policies.

4.1 Representing Assigner's roles

Hospital is not a role in itself, is an instance with extra-semantics to specify that the owner of the policy is not an incontextual URI but one of the type *sch:Hospital*. Although the most straightforward way to represent it is the following:

```
{ ... "permission": [{
  "@type": "odrl:Permission",
  "assigner": "ex:hospital-organization"
  ... }]
```

Listing 1: Excerpt of a policy describing the URI of the hospital without fine-grained semantics

In all the policy formalizations, the authors have chosen to model it as an *odrl:Party* since, according to the class definition in the ODRL ontology, a *odrl:Party* can also be an organisation. Since *odrl:Party* is the *rdfs:subclasssOf* the union of several classes, among which *schema:Organisation*, the final decision to model such assigner is the following:

```
{ ... "permission": [ {
  "@type": "odrl:Permission",
  "assigner": {
    "@type": [ "odrl:Party", "sch:Hospital" ],
    "uid": "ex:hospital-organization",
    "rdfs:description" : "A specific hospital that has adopted this
      policy for access to its data"
  } ... }
```

Listing 2: Excerpt of a policy describing URI of the hospital with fine-grained semantics to represent the concept *sch:Hospital* and the *odrl:Party*

4.2 Roles resolution through evaluation requests & state of the world

Policies 01, 02, 03, 11 and 15 highlight the need to regulate the conduct of certain individuals on the basis of their professional category (e.g. doctors, nurses, etc.), i.e., on the basis of their role within the hospital. In these policies, the notion of group is represented using the class *odrl:PartyCollection* but it cannot be used to create a direct link between a specific user requesting the get access to the data and their role in the hospital. A sample policy in this challenge category is Policy 01 that grants global read access to the patients table to all medical staff.

```
{... "permission": [ ...
  "assignee": {
    "@type": [ "odrl:PartyCollection", "sch:Physician" ],
    "uid": "ex:groups/medical-staff",
    "rdfs:description" : "Anyone who belongs to the medical staff"
  },
  "action": "read",
  "target": "db:PatientsRegistry"
```

```
... }
```

Listing 3: Excerpt of Policy 01 describing the group of physicians as a *odrl:PartyCollection*.

Operational challenge: The challenge is representing semantically meaningful group of people and also, link the identifier of the requester with the party group.

Solution: Taking Policy 01 as an example, the solution was to model in the state of the world the list of those people belonging to the physician staff; although this information is stored in the hospital database. Finally, to check whether the requester is a physician, they must provide their identifier as part of the evaluation request.

```
[{ # Evaluation request
  "@context": [ ... ],
  "@type": "EvaluationRequest",
  "@id": "ex:request/P01-1",
  "evaluatedAction": "odrl:read",
  "evaluatedParty": "ex:id/staff/physician/doctor1", # Provided ID
  "evaluatedTarget": "db:PatientsRegistry" },
{ # State of the world
  "@context" : [ ... ],
  "@id" : "ex:StateP01-1",
  "@type" : "SotW",
  "context" : {
    "@id" : "ex:groups/medical-staff",
    "@type" : "odrl:PartyCollection",
    "rdfs:label" : "Medical Staff",
    "member" : [ {
      "@id" : "ex:id/staff/physician/doctor1", # Registered ID in SoW
      "@type" : ["odrl:Party", "sch:Physician"],
      "odrl:partOf" : {"@id" : "ex:groups/medical-staff"},
      "rdfs:label" : "Dr. Roberto Cheu"
    }, [...] ] }
  ] }
```

Listing 4: Excerpt of the State of the World (SoW) and Evaluation Request representing *odrl:Party*, *odrl:PartyCollection* and requester identity.

Assuming that variables can be used in the definition of policies, an assumption that goes beyond the ODRL standard but is proposed within the ODRE framework [7], another way to approach this issue using the database directly would be to represent the URI of the roles used in the assignee as in Listing 5.

In this case, as mentioned in the previous section, the URI `db:EmployeeRecords/{provided_id}` represents a row in the table of employees. The filter `?status=PHYSICIAN` checks whether the requester that provided the value for the variable `{provided_id}` belongs to MedicalStaff as a physician. Once the variable `provided_id` is instantiated, the existence of such person in the medical staff can be checked, without the need to replicate the information of such tables in the state of the world. This approach, from the practical point of view, might work better because it does not require duplicating information

already contained in the hospital database within the knowledge graph, which could prove useful in addressing certain challenges, as in this case, but it is not always feasible for privacy reasons, as the authors will discuss in the following subsections.

```
{ [..] "permission": [
  {
    "@type": "odrl:Permission",
    "assignee": {
      "@type": [ "odrl:PartyCollection", "sch:Physician" ],
      "uid": "db:EmployeeRecords/{{provided_id}}?status='PHYSICIAN'",
      # {{provided_id}} is a variable
      "rdfs:description" : "Anyone who belongs to the medical staff"
    },
    "action": "read",
    "target": "db:PatientsRegistry"
  ] ] }
```

Listing 5: Excerpt of Policy 01 describing the group of physicians as a `odrl:PartyCollection`, the "uid" is written with ODRE as alternative to Listing 3.

4.3 Representing filtered assets, i.e., policy target

Policy 08 allows researchers to read clinical data for statistical and scientific analysis, but only those records labelled as anonymised. In this case, since the filtering does not depend on a variable, the policy was modelled with a target URI that contains the table filtering, i.e. *db:ClinicalRecords?isAnonymized = 'True'*, instead of having the filtering explicitly written in the constraint of the policy. Similarly, Policy 07 allows emergency physicians to read clinical records marked "CRITICAL" or "EMERGENCY". To address this policy, the target is modelled as *db:ClinicalRecords?patient_status='CRITIC'—patient_status='EMERGENCY'*. Finally, Policy 09 prohibits administrative staff from booking new appointments for patients with outstanding debts, i.e. for those patients whose records are not marked as 'DEBTOR'.

However, in other cases like Policy 04, 05, 06, and 12 this is not enough since there is a linkage between the role (assignee) identity and the table filtering. In Policy 04, patients can read their clinical records from table *db:ClinicalRecords*, i.e., in that table they have their identifier as the foreign key to *db:Patients*. Policy 05 allows physicians to modify the clinical records of those patients that are assigned to them; therefore, the physician identifier appears in the *db:ClinicalRecords* as the foreign key from the table *db:Employees* who are physicians. Policy 12 allows hospital pharmacists to change prescription medications from "PENDING" to "DISPENSED". Finally, Policy 06 allows the head of a specific department to read the clinical record of any patient assigned to a physician of their department.

In the latter cases, since there is a restriction over the table, there is no way in ODRL to express this relationship without the use of variables, or introducing the concept of "policy template".

Operational challenge: Represent a filtered version of the target, with either static values known during the policy definition or with dynamic values unknown before defining the policy, for instance, based on the evaluator identity.

Solution: For static values, an URI strategy can be adopted; however, for dynamic values, ODRL lacks the necessary expressivity to address the problem [8, 7]. To the authors' knowledge, this challenge cannot be solved solely with the ODRL ontology, and due to this reason, the authors decided to follow the ODRE approach [7] introducing the variables in the policy. Taking as an example Policy 04 that is shown below. The values of the patient identity (*ex:request/P04/param-identity*) and their clinical record id (*ex:request/P04/param-record*) are modelled as URIs whose value must be given by the Evaluation Request.

```
{...
  "action": "read",
  "target": "db:ClinicalRecords",
  "constraint": [ {
    "@type": "odrl:Constraint",
    "leftOperand": "ex:request/P04/param-identity",
    "operator": "odrl:isAnyOf",
    "rightOperand": "db:ClinicalRecords?patient_id"
  }, {
    "@type": "odrl:Constraint",
    "leftOperand": "ex:request/P04/param-record",
    "operator": "odrl:isAnyOf",
    "rightOperand": "db:ClinicalRecords?record_id"
  } ... ]}
```

Listing 6: Excerpt of the Policy 04

As can be observed, the target of Policy 04 cannot express that if permission is enforced, patients can only read the clinical records assigned to them. In this case, the identity of the patient and the clinical record itself are represented as part of the evaluation request and bound to a constraint in the policy. However, although this representation may seem to work, is weak since a patient can send a record id that exists and an identity that exists although the record may belong to a different patient id and this policy would allow to read the clinical records (again, any of them since there is no way to restrict the target). Similarly, Policy 09 was modelled as follows:

```
{ ... "target": "db:AppointmentScheduling",
  "constraint": [ {
    "@type": "odrl:Constraint",
    "leftOperand": "ex:request/P09/param-identity",
    "operator": "odrl:eq",
    "rightOperand": "db:BillingInformation?patient_id"
  }, {
    "@type": "odrl:Constraint",
    "leftOperand": "db:BillingInformation?status",
    "operator": "odrl:eq",
    "rightOperand": "DEBTOR"
  } ... ]}
```

Listing 7: Excerpt of the Policy 09

As can be observed, in this case the constraint encodes that the patient exists in the *db:BillingInformation* and the status is *DEBTOR*, however, there is no way to model the relationship between the patient identity and the specific status.

One of the most interesting policies in this category is Policy 14, since it has a constraint that needs to read from the database data (the age) and cannot be duplicated in the state of the world knowledge graph. Also, it represents that the requester (the legal guardian) exists in the database, and that the under aged patient as well. However, there is no manner to relate both three values, meaning that the same person is under age, is a patient, has a legal guardian, and the requester id matches one of his/her legal guardians.

```
{ ...
  "action": "read",
  "target": "db:ClinicalRecords",
  "constraint": [ {
    "@type": "odrl:Constraint",
    "leftOperand": "ex:request/P04/param-identity",
    "operator": "odrl:isAnyOf",
    "rightOperand": "db:ClinicalRecords?patient_id"
  }, {
    "@type": "odrl:Constraint",
    "leftOperand": "ex:request/P04/param-record",
    "operator": "odrl:isAnyOf",
    "rightOperand": "db:ClinicalRecords?record_id"
  } ] ... } }
```

Listing 8: Excerpt of the Policy 14

4.4 Dynamic state of the world based on identity

Policy 10 states that a person who must be a nurse and, therefore, is part of the party collection can (*odrl:permission*) read or modify (*odrl:read* or *odrl:modify*, respectively) if that person is within their shift. This policy has two main issues, on the one hand, the shift of each nurse changes depending on their identity so one policy cannot contain all the shifts associated with each nurse. However, assuming that there is one policy per shift, the ODRL policy itself does not have a reference to the identity of the specific nurse, since the policy applies to the party collection of the nurse staff. In order to express this policy uniquely with ODRL, it would be necessary to remove the party collection and state the specific shift per nurse using their identifier; which gives away sensitive information hardcoding that information in the policy. Also, another documented issue is that ODRL does not allow time to be represented without a date as an operand [5], and for this example the shift (right operand) cannot be expressed using *odrl:dateTime*, it would be more suitable to include something that expresses the time interval alongside the day of the week.

Operational challenge: Link data contained in a protected data source (the shifts table), which should not appear in the state of the world due to privacy reasons, and the assignee identity who belongs to a group of people (the nursing staff). That link holds the necessary information required to write the policy; however, such a link cannot appear in the policy.

Solution: It is necessary to introduce the concept of policy templates that may contain unknown variables prior to the policy evaluation and that are instantiated with the evaluation

request information producing a fully written ODRL policy on the fly. This problem and solution has already been introduced into the state of the art by other research articles [8, 7]. In order to specify Policy 10 authors decided to follow the ODRE approach [7] introducing variables in the policy.

```
{ ... "permission": [{
  "@type": "odrl:Permission",
  "assigner": {
    "@type": [ "odrl:Party", "sch:Hospital"],
    "uid": "ex:hospital-organization",
    "rdfs:description" : "A specific hospital that owns this policy"
  },
  "assignee": {
    "@type": [ "odrl:PartyCollection", "coral:NurseStaff"
  ],
    "uid": "ex:groups/nursing-staff",
    "rdfs:description" : "Anyone who belongs to the nursing staff"
  },
  "action": [ "read", "modify" ],
  "target": "db:MedicationPrescriptions",
  "constraint": [{
    "@type": "odrl:Constraint",
    "leftOperand": "dateTime",
    "operator": "odrl:gteq",
    "rightOperand": "ex:subjectShiftStart"
  }, {
    "@type": "odrl:Constraint",
    "leftOperand": "dateTime",
    "operator": "odrl:lteq",
    "rightOperand": "ex:subjectShiftEnd"
  } ] ... } ] }
```

Listing 9: Excerpt of the Policy 10

It can be observed that Policy 10 requires two values, which are the shift start and the shift ends, that must be bound somehow to the identity of a nurse. In this policy, as a solution, it has been assumed that the *ex:subjectShiftStart* and the *ex:subjectShiftEnd* are URIs that relate to those values in the State of the World as shown below.

```
{ ... "@type" : "SotW",
  "context" : {
    "@id" : "ex:groups/nursing-staff",
    "@type" : "odrl:PartyCollection",
    "rdfs:label" : "Nursing Staff",
    "member" : [ {
      "@id" : "ex:id/staff/nurse/nurse1",
      "@type" : ["odrl:Party", "coral:NurseStaff"],
      "odrl:partOf" : {"@id" : "ex:groups/nursing-staff"},
      "rdfs:label" : "Iker Scrogam"
    }, {
```

```

    "@id" : "ex:subjectShiftStart",
    "@type" : "coral:WorkingShiftStarts",
    "coral:value" : "09:00",
    "coral:shiftAppliesTo" : "ex:id/staff/nurse/nurse1"
  },{
    "@id" : "ex:subjectShiftEnd",
    "@type" : "coral:WorkingShiftEnds",
    "coral:value" : "18:00",
    "coral:shiftAppliesTo" : "ex:id/staff/nurse/nurse1"
  } ...

```

Listing 10: Excerpt of the State of the World for Policy 10

However, this information, which under this approach is already present in the database, should be duplicated in the state of the world. This leads to data duplication and potential privacy leakage. Furthermore, it requires to express for each hospital's employee all their shifts.

4.5 Policy expiration

In Policy 13, a physician allows a third-party physician, who is an external specialist, to access a patient record. In this scenario, a physician already has a policy that allows them to read the records, and due to medical reasons, a new policy is generated for the specialist with a time to live after which it expires.

Operational challenge: Generate a policy that has temporal validity after which it expires. Also, the policy needs to represent either the person, or the role, that grants the temporal access to a third party and such third party.

Solution: Since there is no concept similar to *deadline* or *time to live* in the ODRL ontology, authors have chosen to represent this restriction as a constraint based on the date time for the left operand and calculating the expiration date written in the right operand as a constant value. However, an interesting approach would be to calculate the time elapsed since the policy was issued. In the policy written, the person who grants the access does not appear explicitly and only the third-party is shown. To showcase this operational challenge and the proposed solution, Policy 13 is shown in the following listing.

```

{ ...
  "permission": [
    {
      "@type": "odrl:Permission",
      "assigner": { .... },
      "assignee": { ... },
      "action": "read",
      "target": "db:ClinicalRecords/324145?record_id='AB12'",
      "rdfs:label": "The target specifies the patient (324145) and the
        specific clinical record (AB12) that can be accessed by the
        specialist",
      "constraint": [{
        "leftOperand": "dateTime",
        "operator": "odrl:lt",
        "rightOperand": { "@value": "2026-05-04T13:51:59",

```

```

    "type": "xsd:datetime"}
  } ... }

```

Listing 11: Excerpt of the Policy 13

As it can be observed, the concept of temporal validity is modelled by the constraint that checks whether the current date and time is less than another. In a real-world scenario, the physician must assess whether the policy should cease to be valid and state this explicitly, as set out in Policy 13. Nevertheless, the formalization of the policy does not entail the meaning conveyed in the scenario where the policy has a "time to live". As a solution, Policy 13 may work however, from the semantic point of view of the ontology, the terms needed to express the restriction of the scenario are missing.

5 Conclusions and Future work

This article presented a curated dataset of 15 hospital access control policies expressed in natural language, RLS, ODRL, and ODRE syntax. The policy corpus captures a diverse set of constraint patterns, including identity-based restrictions, organisational roles, temporal activation, and resource-state conditions. Although these examples illustrate the expressive capabilities of ODRL for modelling complex access control scenarios, they also reveal numerous limitations. In particular, the analysis shows that certain constraints cannot be adequately expressed using the current ODRL model, especially when policies depend on relational dependencies, filtered resources, or contextual information derived from protected data sources. These limitations highlight the need to introduce policy templates and variables that can be instantiated at evaluation time. Such mechanisms enable the representation of dynamic policy parameters while avoiding the replication of sensitive operational data in the state of the world.

The resulting dataset provides a reusable resource for evaluating ODRL-based policy engines and studying the translation of native database access control rules into semantic policy representations. Future work will focus on extending the dataset with additional policies, supporting automated consistency validation, and investigating template-based policy modelling mechanisms for usage control.

Acknowledgments

This work has been partially supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement 2023-2026 with the Universidad Politécnica de Madrid in Line A, Emerging PhD researchers through the project GUIA (M230020126A-AJCA). Also, the work has been partially supported by MALTA, PID2024-159504OB-I00 funded by MICIU/AEI/10.13039/501100011033.

References

- [1] Shayan Ahmadi, Mohammad Nassiri, and Mohsen Rezvani. Xacbench: a xacml policy benchmark. *Soft Computing*, 24(21):16081–16096, 2020.
- [2] Ines Akaichi and Sabrina Kirrane. A comprehensive review of usage control frameworks. *Computer Science Review*, 56:100698, 2025.

- [3] Elisa Bertino and Ravi Sandhu. Database security-concepts, approaches, and challenges. *IEEE Transactions on Dependable and secure computing*, 2(1):2–19, 2005.
- [4] Piero Andrea Bonatti, Nicoletta Fornara, and Andreas Harth. Towards a Formal Semantics of the Open Digital Rights Language (ODRL 2.2). In *ODRL and Beyond: Practical Applications and Challenges for Policy-based Access and Usage Control (OPAL 2025)*, co-located with the *Extended Semantic Web Conference*, volume Vol-3977, 2025.
- [5] Juan Cano-Benito, Andrea Cimmino, and Raúl García-Castro. Towards time privacy policies in ODRL. In *Proceedings of the 1st NeXt-generation Data Governance Workshop 2024 co-located with 20th International Conference on Semantic Systems (SEMANTiCS 2024)*, volume Vol-3891, 2024.
- [6] Andrea Cimmino, Juan Cano-Benito, and Raúl García-Castro. Practical challenges of odrl and potential courses of action. In *Companion Proceedings of the ACM Web Conference 2023*, pages 1428–1431, 2023.
- [7] Andrea Cimmino, Juan Cano-Benito, and Raúl García-Castro. Open Digital Rights Enforcement framework (ODRE): From descriptive to enforceable policies. *Computers & Security*, 150:104282, 2025.
- [8] Andrea Cimmino and Nicoletta Fornara. Improving ODRL 2.2: current limitations and theoretical solutions. In *ODRL and Beyond: Practical Applications and Challenges for Policy-based Access and Usage Control (OPAL 2025)*, co-located with the *Extended Semantic Web Conference*, volume Vol-3977, 2025.
- [9] Marina De Vos, Sabrina Kirrane, Julian A. Padget, and Ken Satoh. ODRL Policy Modelling and Compliance Checking. In *Rules and Reasoning (RuleML+RR 2019)*, volume 11784 of *Lecture Notes in Computer Science*, pages 36–51. Springer, 2019.
- [10] Xueying Deng, Matthias De Geyter, Bart Matthys, Casper Van Gheluwe, David Vermeir, and Matthias Stevens. Flemish health data space implementation: Technical overview and challenges. In *The Third International Workshop on Semantics in Dataspaces*, co-located with the *Extended Semantic Web Conference*, 2025.
- [11] Korrry Douglas and Susan Douglas. *PostgreSQL: a comprehensive guide to building, programming, and administering PostgreSQL databases*. SAMS publishing, 2003.
- [12] William Eiers, Ganesh Sankaran, Albert Li, Emily O’Mahony, Benjamin Prince, and Tevfik Bultan. Quacky: Quantitative access control permissiveness analyzer. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–5, 2022.
- [13] William Eiers, Ganesh Sankaran, Albert Li, Emily O’Mahony, Benjamin Prince, and Tevfik Bultan. Quantifying permissiveness of access control policies. In *Proceedings of the 44th International Conference on Software Engineering*, pages 1805–1817, 2022.
- [14] Beatriz Esteves and Harshvardhan J. Pandit. duo-odrl-dpv: artefacts for DUO-ODRL-DPV alignment and policy patterns. <https://github.com/besteves4/duo-odrl-dpv>, 2024.
- [15] Beatriz Esteves, Harshvardhan J Pandit, and Víctor Rodríguez-Doncel. ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 298–306. IEEE, 2021.
- [16] Beatriz Esteves, Harshvardhan J Pandit, and Víctor Rodríguez-Doncel. Odrl profile for expressing consent through granular access control policies in solid. In *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 298–306. IEEE, 2021.
- [17] Beatriz Esteves, Harshvardhan J. Pandit, and Víctor Rodríguez-Doncel. ODRL Profile for Access Control (OAC). <https://besteves4.github.io/odrl-access-control-profile/oac.html>, 2023.
- [18] Beatriz Esteves, Víctor Rodríguez-Doncel, Harshvardhan J Pandit, Nicolas Mondada, and Pat McBennett. Using the odrl profile for access control for solid pod resource governance. In *European Semantic Web Conference*, pages 16–20. Springer, 2022.
- [19] Beatriz Esteves, Wout Slabbinck, Yassir Sellami, Andrea Cimmino, Víctor Rodríguez-Doncel, and

- Ruben Verborgh. Capturing Requests and Context for ODRL-based Access and Usage Control. In *Joint Proceedings of the 16th Workshop on Ontology Design and Patterns and the 1st Workshop on Bridging Hybrid Intelligence and the Semantic Web (WOP-HAIBRIDGE 2025) colocated with the 24th International Semantic Web Conference (ISWC 2025)*, 2025.
- [20] Nicoletta Fornara, Víctor Rodríguez-Doncel, Beatriz Esteves, Simon Steyskal, Benedict Whittam-Smith, Yassir Sellami, and Andrea Cimmino. ODRL Formal Semantics (Draft). <https://w3c.github.io/odrl/formal-semantics/>, 2026.
- [21] Arghavan Hosseinzadeh, Andreas Eitel, and Christian Jung. A systematic approach toward extracting technically enforceable policies from data usage control requirements. In *ICISSP*, pages 397–405, 2020.
- [22] Renato Iannella, Martin Steidl, Stuart Myles, and Víctor Rodríguez-Doncel. ODRL Vocabulary & Expression 2.2. W3C Recommendation, 2018.
- [23] Renato Iannella and Serena Villata. ODRL Information Model 2.2. W3C Recommendation, 2018.
- [24] International Data Spaces Association. *International Data Spaces Reference Architecture Model Version 3.0*. International Data Spaces Association, 2019.
- [25] International Data Spaces Association. Technical agreements of the international data spaces association. <https://docs.internationaldataspaces.org/ids-knowledgebase/idsa-rulebook/>, 2024.
- [26] Markus Lorch, Seth Proctor, Rebekah Lepro, Dennis Kafura, and Sumit Shah. First experiences using XACML for access control in distributed systems. In *Proceedings of the 2003 ACM workshop on XML security*, pages 25–37, 2003.
- [27] Daham M Mustafa, Abhishek Nadgeri, Diego Collarana, Benedikt T Arnold, Christoph Quix, Christoph Lange, and Stefan Decker. From instructions to ODRL usage policies: An ontology guided approach. *arXiv preprint arXiv:2506.03301*, 2025.
- [28] Harshvardhan J Pandit and Beatriz Esteves. Enhancing data use ontology (DUO) for health-data sharing by extending it with ODRL and DPV. *Semantic Web*, 15(4):1473–1498, 2024.
- [29] Malo Revel, Aurélien Lamerclerie, Annie Foret, and Zoltan Miklos. Extracting ODRL digital right representations from license texts using AMR. In *ASAIL 2023-Automated Semantic Analysis of Information in Legal Text*, pages 1–7, 2023.
- [30] Víctor Rodríguez-Doncel, Serena Villata, and Asunción Gómez-Pérez. A dataset of RDF licenses. In Rinke Hoekstra, editor, *Legal Knowledge and Information Systems - JURIX 2014: The Twenty-Seventh Annual Conference, Jagiellonian University, Krakow, Poland, 10-12 December 2014*, volume 271 of *Frontiers in Artificial Intelligence and Applications*, pages 187–188. IOS Press, 2014.
- [31] Raphael Scheible, Fabian Thomczyk, Marco Blum, Micha Rautenberg, Andrea Prunotto, Suhail Yazijy, and Martin Boeker. Integrating row level security in i2b2: segregation of medical records into data marts without data replication and synchronization. *JAMIA open*, 6(3):ooad068, 2023.
- [32] Wout Slabbinck, Julián Rojas Meléndez, Beatriz Esteves, Pieter Colpaert, and Ruben Verborgh. Interoperable Interpretation and Evaluation of ODRL Policies. In *The Semantic Web*, pages 192–209, 2025.
- [33] Simon Steyskal and Axel Polleres. Defining expressive access policies for linked data using the ODRL ontology 2.0. In *Proceedings of the 10th International Conference on Semantic Systems*, pages 20–23, 2014.
- [34] Simon Steyskal and Axel Polleres. Towards formal semantics for ODRL policies. In *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, pages 360–375. Springer, 2015.
- [35] Adarsh Vatsa, Bethel Hall, and William Eiers. Exploring Large Language Models for Access Control Policy Synthesis and Summarization, 2025.
- [36] Caroline Winslow and Lydia M Falk. Row-level security enforcement strengths in apex multi-tenant deployments. *Journal of Artificial Intelligence in Fluid Dynamics*, 1(2):7–12, 2022.

- [37] Tony W York and Don MacAlister. *Hospital and healthcare security*. Butterworth-Heinemann, 2015.
- [38] Wanle Zhong, Keman Huang, and Xiaoyong Du. AgentODRL: A Large Language Model-based Multi-agent System for ODRL Generation, 2025.