

# Chapter 1: ODRL

## 1.1 Il linguaggio

### 1.1.1 Definizione ed obiettivi

“L’ Open Digital Rights Language (ODRL) è un linguaggio per l’espressione di policy che definisce: un modello dell’informazione flessibile ed interoperativo, un vocabolario e un meccanismo di codifica per la rappresentazione delle istruzioni sull’uso di contenuti o servizi”[4].

Il linguaggio si pone all’interno dello scenario applicativo nel quale vi è la necessità di definire:

- quali azioni siano permesse o proibite su una risorsa. Queste regole possono essere imposte da leggi o direttamente dal possessore dell’asset o servizio;
- indicare quali attori interagiscono con le policy definite; in particolare chi può definire le policy e a chi si applicano;
- indicare eventuali limiti riguardanti i permessi ed i divieti espressi;.

Avere un modello standard per definire questi bisogni dà 2 fondamentali vantaggi:

- chi possiede l’asset è in grado di definire in modo chiaro quali siano le azioni che un consumatore può fare, evitando quindi usi indesiderati;
- chi usufruisce dell’asset conosce in modo preciso quali azioni può compiere, evitando così di infrangere regole o leggi.

ODRL definisce un modello semantico di permessi, divieti ed obblighi, che può essere usato per descrivere le modalità d’uso di un contenuto. In particolare si cerca di definire i concetti chiave per la creazione di policy machine-readable collegate direttamente all’asset al quale sono associate, permettendo all’utente finale di reperire facilmente informazioni sulla risorsa che utilizza. Quest’ultimo requisito è soddisfatto, poiché ODRL è costruito seguendo i *Linked Data principles*[1]:

- Utilizzo di URIs come nomi per le risorse;
- Gli URI sono indirizzi HTTP in modo che le persone possano cercare informazioni sulle risorse;
- L'URI deve fornire informazioni utili sulla risorsa;
- Tra le informazioni, fornire altri URI, in modo che l'utente possa raggiungere altre informazioni.

Nonostante questi principi siano più indicati per un'implementazione graph-based, è possibile anche definire utilizzi che non tengano conto dei link tra le varie informazioni.

### 1.1.2 Il modello

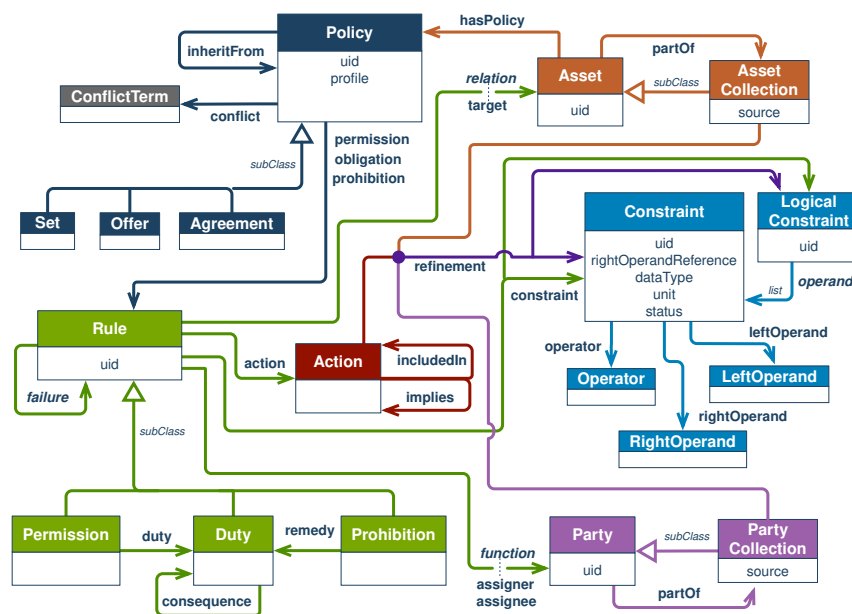


Figure 1.1: Schema del modello ODRL[4]

Come visibile all'interno dello schema in figura 1.1, il modello è basato sulle seguenti entità principali:

- **Policy**: un gruppo di una o più regole;
- **Regola**: concetto astratto che racchiude le caratteristiche comuni di **permesso**, **divieto**, **doveri**;
- **Asset**: risorsa o collezione di risorse soggette a regole;

- **Azione:** operazione su un asset;
- **Party:** entità o insieme di entità con un certo ruolo in una regola;
- **Limiti:** espressione logica o booleana imposta su azioni, party, asset o regole.

## Vocabolari

“L’ *ODRL Vocabulary and Expression* descrive i termini usati dalle policy ODRL e come codificarle”[3]. All’interno di ODRL, i vocabolari utilizzati per definire i termini all’interno delle policy vengono detti **profili**, i quali possono essere usati per definire termini che supportano specifiche applicazioni; all’interno di un profilo è possibile, ad esempio, fornire le specifiche riguardanti nuove sottoclassi di termini già presenti nei vocabolari standard di ODRL. I 2 vocabolari principali definiti sono:

- **ODRL Core Vocabulary:** rappresenta la minima espressione di policy supportata;
- **ODRL Common Vocabulary:** arricchisce il vocabolario precedente con un gruppo di azioni generiche, nuove sottoclassi per le policy, ruoli per i party e relazioni tra gli asset.

Una delle principali differenze tra i due vocabolari, la si ha all’interno delle **azioni** che possono essere indicate: nel primo caso si hanno a disposizione solamente 2 azioni **use** e **transfer**, nel secondo caso queste 2 azioni vengono estese da diverse azioni figlie, come mostrato in figura 1.2

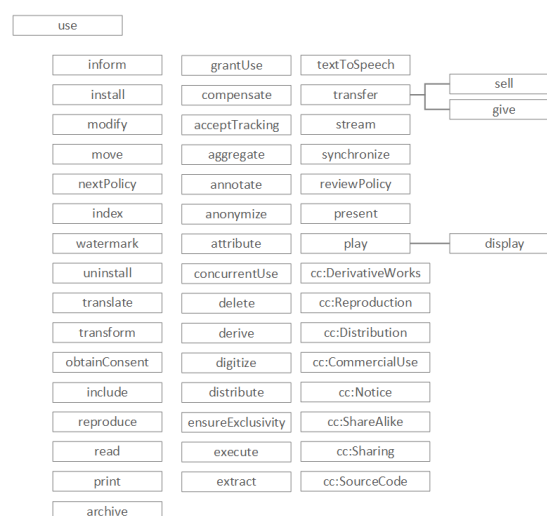


Figure 1.2: Tutte le azioni mostrate sono figlie di use, ad eccezioni di transfer e le sue sottoazioni[5]

## Policy

Come definito nel modello presente nella sezione 1.1.2, una policy è un gruppo non vuoto di **regole** e, quindi, di **permessi**, **divieti** o **obblighi**. Una policy deve soddisfare i seguenti requisiti:

- deve avere un identificativo univoco, detto **uid**;
- deve avere almeno una regola;
- può specificare un profilo, obbligatorio se non si usa il Core Vocabulary mostrato nella sezione 1.1.2;
- può specificare una policy da cui eredita le proprietà;
- può specificare una strategia per la risoluzione dei conflitti.

Come visibile dalla figura 1.1, una policy ha 3 possibili sottoclassi:

- **Set**: un insieme di regole che hanno effetto;
- **Agreement**: regole concesse ad una entità assegnataria da una assegnatrice;
- **Offer**: proposta di una regola da parte di un assegnatore.

Di seguito un esempio di policy definita mediante ODRL:

```
1 {  
2   "@context": "http://www.w3.org/ns/odrl.jsonld",  
3   "@type": "Set",  
4   "uid": "http://example.com/policy:1010",  
5   "permission": [{  
6     "target": "http://example.com/asset:9898.movie",  
7     "action": "use"  
8   }]  
9 }
```

Listing 1.1: Policy con sottoclasse **Set**

La policy mostrata nel listing 1.1 presenta i campi:

- **@type**: serve ad indicarne la sottoclasse;
- **@context**: serve ad indicare che il file deve essere conforme ad ODRL, rappresentato dall'URL da cui si può ottenere l'ODRL Common Vocabulary[2];

- nel contesto sono presenti altri link per le altre proprietà, ad esempio quello per la descrizione di *Set* e per *use*;
- non usando termini fuori dai 2 vocabolari principali, non necessita la definizione di un profilo;
- l'id univoco è rappresentato da un URL che porta alle informazioni relative la risorsa.

## Asset

Come definito nel modello presente nella sezione 1.1.2, un asset è una risorsa o una collezione di risorse soggette a regole. Un asset può essere una qualunque risorsa identificabile. Ha **AssetCollection** come sottoclasse, la quale rappresenta una collezione di asset. La classe asset può avere:

- un identificativo univoco, il quale può essere omesso se l'asset è fornito direttamente con la policy; le specifiche di ODRL, pur supportando questo caso d'uso, sconsigliano questa pratica;
- una o più proprietà denominate **partOf**: identifica le collezioni di cui fa parte l'asset, il quale può essere a sua volta una collezione.

Esempio di utilizzo in una regola:

```

1 {
2   "@context": "http://www.w3.org/ns/odrl.jsonld",
3   "@type": "Offer",
4   "uid": "http://example.com/policy:3333",
5   "profile": "http://example.com/odrl:profile:02",
6   "permission": [{
7     "target": "http://example.com/asset:3333",
8     "action": "display",
9     "assigner": "http://example.com/party:0001"
10  }]
11 }
```

Listing 1.2: Utilizzo di asset nella proprietà **target** di una regola

La sottoclasse **AssetCollection** può avere i seguenti 2 campi aggiuntivi:

- **source**: sostituisce il campo **uid** nelle classe **AssetCollection** all'interno di un **refinement**;

- una o più **refinement**: limiti riguardanti la collezione che identificano solamente un sottogruppo di asset al suo interno.

Esempio di utilizzo della proprietà **partOf**:

```

1 {
2     "@type": "dc:Document",
3     "@id": "http://example.com/asset:111.doc",
4     "dc:title": "Annual Report",
5     ...
6     "odrl:partOf": "http://example.com/archive1011",
7     ...
8 }
```

Listing 1.3: L'asset definito è parte del target presente nel listing 1.2

In quest'ultimo esempio si ha che l'asset con id "http://example.com/asset:111.doc" è definito come parte della collezione "http://example.com/archive1011". Questa definizione ha come effetto l'applicarsi della policy presente nel listing 1.2 anche all'asset che fa parte della collezione.

## Party

Come definito nel modello presente nella sezione 1.1.2, un party è una entità o una collezione di entità con una funzione determinata in una regola. Un party può essere un qualunque soggetto con un ruolo attivo nelle regole o che produce un effetto specifico, ad esempio controlla che le azione relative ad un dovere vengano effettuate. Ha **PartyCollection** come sottoclasse, la quale rappresenta una collezione di entità. La classe party può avere:

- un identificativo univoco, il quale può essere omesso se è possibile definire in altro modo l'entità; le specifiche di ODRL, pur supportando questo caso d'uso, sconsigliano questa pratica;
- una o più proprietà denominate **partOf**: identifica le collezioni di cui fa parte l'entità, la quale può essere a sua volta una collezione.

Esempio di utilizzo in una regola:

```

1 {
2     "@context": "http://www.w3.org/ns/odrl.jsonld",
3     "@type": "Agreement",
4     "uid": "http://example.com/policy:8888",
```

```

5 "profile": "http://example.com/odrl:profile:04",
6 "permission": [{
7   "target": "http://example.com/music/1999.mp3",
8   "assigner": "http://example.com/org/sony-music",
9   "assignee": "http://example.com/people/billie",
10  "action": "play"
11 }]
12 }

```

Listing 1.4: Utilizzo di party nelle proprietà **assigner** ed **assignee** di una regola

La sottoclasse **PartyCollection** può avere i seguenti 2 campi aggiuntivi:

- **source**: sostituisce il campo **uid** nella classe **PartyCollection** all'interno di un **refinement**;
- una o più **refinement**: limiti riguardanti la collezione che identificano solamente un sottogruppo di asset al suo interno.

Esempio di utilizzo della proprietà **partOf**:

```

1 {
2   "@type": "vcard:Individual",
3   "@id": "http://example.com/person/murphy",
4   "vcard:fn": "Murphy",
5   "vcard:hasEmail": "murphy@example.com",
6   ...
7   "odrl:partOf": "http://example.com/team/A",
8   ...
9 }

```

Listing 1.5: L'entità definita è parte di una **PartyCollection**

In quest'ultimo esempio si ha che l'entità con id "http://example.com/person/murphy" è definita come parte della collezione "http://example.com/team/A". Questa definizione ha come effetto l'affidare le funzioni di quest'ultima collezione anche alla singola entità che ne fa parte.

## Action

Come definito nel modello presente nella sezione 1.1.2, **action** è una classe che rappresenta un'operazione che può essere esercitata su un asset, al quale viene associata mediante la proprietà **action** di una regola. Nell' ODRL Core Vocabulary sono presenti 2 azioni principali:

- use: un qualsiasi utilizzo dell'asset;
- transfer: una qualsiasi azione che preveda il trasferimento di proprietà dell'asset;

Un'azione può avere le seguenti proprietà:

- refinement: raffinamenti semantici sull'azione, come ad esempio l'ammontare di un pagamento, il luogo nel quale l'azione può essere eseguita o il tempo massimo di esecuzione;
- includedIn: esprime l'azione padre; la conseguenza di questa dichiarazione risulta essere che tutte le regole applicate all'azione padre, devono valere anche per l'azione figlia;
- implies: esprime un'azione che non deve essere vietata per permettere l'azione con questa proprietà, ma le 2 azioni non hanno una relazione espressa tramite *includedIn*<sup>1</sup>.

Come anticipato nel paragrafo 1.1.2 relativo ai profili, l'ORDL Common Vocabulary utilizza la proprietà *includedIn* per aggiungere azioni figlie sia ad *use* che *transfer*, come già mostrato nella figura 1.2.

Di seguito un esempio di azione in una regola:

```

1
2 {
3     "@context": "http://www.w3.org/ns/odrl.jsonld",
4     "@type": "Offer",
5     "uid": "http://example.com/policy:1012",
6     "profile": "http://example.com/odrl:profile:06",
7     "permission": [{
8         "target": "http://example.com/music:1012"
9         ,
10        "assigner": "http://example.com/org:abc",
11        "action": "play"
12    }]
13 }
```

Listing 1.6: L'azione **play** è presente nella proprietà **action** della regola

<sup>1</sup>attualmente né l'ODRL Core Vocabulary né l'ODRL Common Vocabulary presentano azioni con questa proprietà



## Constraint e Logical Constraint

Come definito nel modello presente nella sezione 1.1.2, **constraint** è una classe usata per comparare 2 espressioni che non sono constraint a loro volta, utilizzando un operatore relazionale. Rappresentano una limitazione tramite un confronto, la quale può essere soddisfatta o non soddisfatta. La classe presenta le seguenti proprietà:

- uno o nessun identificativo univoco, qualora si volesse riutilizzare l'espressione di confronto definita;
- un **leftOperand**: elemento a sinistra dell'operatore di confronto;
- un sottotipo **operator**: operatore di confronto;
- uno tra:
  - **rightOperand**: elemento a destra dell'operatore di confronto, identificato direttamente;
  - **rightOperand**: elemento a destra dell'operatore di confronto, identificato con un riferimento;
- uno o nessun **dataType**: definisce il tipo dell'operando di destra;
- una o nessuna **unit**: unità di misura dell'operando di destra;
- uno o nessun **status**: per l'elemento di sinistra.

Oltre ai normali constraint, il modello definisce anche dei logical constraint, ovvero operazioni logiche su altri constraint già definiti. In questo caso le proprietà sono:

- uno o nessun identificativo univoco, qualora si volesse riutilizzare l'espressione di confronto definita;
- un sottotipo di **operand**: operatore logico tra i constraint espressi come lista al suo interno.

Esempi di utilizzi dei constraint:

```
1 {  
2   "@context": "http://www.w3.org/ns/odrl.jsonld",  
3   "@type": "Offer",  
4   "uid": "http://example.com/policy:6161",  
5   "profile": "http://example.com/odrl:profile:10",  
6   "permission": [{  
7     "target": "http://example.com/document:1234",
```

```

8   "assigner": "http://example.com/org:616",
9   "action": [{
10      "rdf:value": { "@id": "odrl:print" },
11      "refinement": [{
12         "leftOperand": "resolution",
13         "operator": "lteq",
14         "rightOperand": { "@value": "1200", "
15            @type": "xsd:integer" },
16         "unit": "http://dbpedia.org/resource/
17            Dots_per_inch"
18      }]
19   }]

```

Listing 1.7: Constraint su azione: l'azione **print** è permessa solo per risoluzioni minori di 1200 dpi

```

1  {
2   "@context": "http://www.w3.org/ns/odrl.jsonld",
3   "@type": "Offer",
4   "uid": "http://example.com/policy:88",
5   "profile": "http://example.com/odrl:profile:10",
6   "permission": [{
7      "target": "http://example.com/book/1999",
8      "assigner": "http://example.com/org/paisley-park",
9      "action": [{
10         "rdf:value": { "@id": "odrl:reproduce" },
11         "refinement": {
12            "xone": {
13               "@list": [
14                  { "@id": "http://example.com/p:88/C1" },
15                  { "@id": "http://example.com/p:88/C2" }
16               ]
17            }
18         }
19      }]
20   }]

```

Listing 1.8: Constraint logico su azione: l'azione **reproduce** è permessa solo nella forma di uno dei due constraint listati

```
1 {
2   "@context": "http://www.w3.org/ns/odrl.jsonld",
3   "@type": "Offer",
4   "uid": "http://example.com/policy:4444",
5   "profile": "http://example.com/odrl:profile:11",
6   "permission": [{
7     "assigner": "http://example.com/org88",
8     "target": {
9       "@type": "AssetCollection",
10      "source": "http://example.com/media-catalogue",
11      "refinement": [{
12        "leftOperand": "runningTime",
13        "operator": "lt",
14        "rightOperand": {
15          "@value": "60",
16          "@type": "xsd:integer"
17        },
18        "unit": "http://qudt.org/vocab/unit/MinuteTime"
19      }]
20    },
21    "action": "play"
22  }]
23 }
```

Listing 1.9: Constraint su asset: l'azione **play** è permessa solo sui target di durata strettamente inferiore a 60 minuti

```
1 {
2   "@context": "http://www.w3.org/ns/odrl.jsonld",
3   "@type": "Agreement",
4   "uid": "http://example.com/policy:4444",
5   "profile": "http://example.com/odrl:profile:12",
6   "permission": [{
7     "target": "http://example.com/myPhotos:BdayParty",
```

```

8  "assigner": "http://example.com/user44",
9  "assignee": {
10     "@type": "PartyCollection",
11     "source": "http://example.com/user44/friends",
12     "refinement": [{
13         "leftOperand": "foaf:age",
14         "operator": "gt",
15         "rightOperand": {
16             "@value": "17",
17             "@type": "xsd:integer"
18         }
19     }]
20 },
21 "action": { "@id": "ex:view" }
22 ]]
23 }

```

Listing 1.10: Constraint su party: l'azione **view** è permessa solo alle entità con età strettamente superiore a 17 anni

## Rule

Come definito nel modello presente nella sezione 1.1.2, **rule** è una classe astratta che raccoglie gli aspetti comuni della classi **permission**, **prohibition**, and **duty**. Rappresenta una delle regole all'interno della policy. Presenta le seguenti proprietà:

- una **action**: azione regolamentata;
- una o nessuna **relation**: asset sul quale si applica la regola;
- una, nessuna o più **function**: funzioni che un party può avere all'interno di una regola;
- uno, nessuno o più **constraint** : limiti applicati alla validità della regola;
- uno o nessun identificativo univoco, necessario solo qualora si usare la regola per ereditarne le proprietà;

Le sottoclassi sono così definite:

- **permission**: permette un'azione sull'asset specificato, con tutti i **refinement** di quest'ultima soddisfatti; inoltre l'azione può essere eseguita solo se tutti i limiti

della regola sono soddisfatti e ogni dovere espresso come **duty** è stato rispettato. Un permesso rende obbligatoria la **relation** denominata **target**;

- **prohibition**: vieta un'azione sull'asset specificato, con tutti i **refinement** di quest'ultima soddisfatti; inoltre l'azione non può essere eseguita solo se tutti i limiti della regola sono soddisfatti; se si infrange il divieto, ogni dovere espresso come **remedy** deve essere eseguito. Un divieto rende obbligatoria la **relation** denominata **target**;
- **duty**: obbligo di eseguire un'azione, con tutti i **refinement** di quest'ultima soddisfatti; un dovere è compiuto se tutti i suoi limiti sono soddisfatti e la sua azione effettuata, con tutti i **refinement** definiti. Se un dovere non è stato compiuto, bisogna eseguirne le **consequences**, ovvero altri doveri da compiere.

Esempi di regole all'interno di policy:

```
1 {
2   "@context": "http://www.w3.org/ns/odrl.jsonld",
3   "@type": "Offer",
4   "uid": "http://example.com/policy:9090",
5   "profile": "http://example.com/odrl:profile:07",
6   "permission": [{
7     "target": "http://example.com/game:9090",
8     "assigner": "http://example.com/org:xyz",
9     "action": "play",
10    "constraint": [{
11      "leftOperand": "dateTime",
12      "operator": "lteq",
13      "rightOperand": { "@value": "2017-12-31", "@type":
14        : "xsd:date" }
15    }]
16  }]
```

Listing 1.11: La regola esprime il permesso di eseguire l'azione **play** sul target fino al giorno 2017-12-31 compreso

```
1 {
2   "@context": "http://www.w3.org/ns/odrl.jsonld",
3   "@type": "Agreement",
4   "uid": "http://example.com/policy:5555",
```

```

5  "profile": "http://example.com/odrl:profile:08",
6  "prohibition": [{
7      "target": "http://example.com/photoAlbum:55",
8      "action": "archive",
9      "assigner": "http://example.com/MyPix:55",
10     "assignee": "http://example.com/assignee:55"
11  }]
12 }

```

Listing 1.12: La regola esprime il divieto di eseguire l'azione **archive** sul target

```

1  {
2  "@context": "http://www.w3.org/ns/odrl.jsonld",
3  "@type": "Agreement",
4  "uid": "http://example.com/policy:42",
5  "profile": "http://example.com/odrl:profile:09",
6  "obligation": [{
7      "assigner": "http://example.com/org:43",
8      "assignee": "http://example.com/person:44",
9      "action": [{
10         "rdf:value": {
11             "@id": "odrl:compensate"
12         },
13         "refinement": [
14             {
15                 "leftOperand": "payAmount",
16                 "operator": "eq",
17                 "rightOperand": { "@value": "500.00", "@type": "xsd:
18                     decimal" },
19                 "unit": "http://dbpedia.org/resource/Euro"
20             }
21         ]
22     }]
23 }

```

Listing 1.13: La regola esprime l'obbligo di eseguire l'azione **compensate**, specificando come **refinement** l'ammontare del pagamento

# Bibliography

- [1] Chris Bizer. Linkeddata.
- [2] Renato Iannell, Michael Steidl, Stuart Myles, and Víctor Rodríguez-Doncel. OdrI version 2.2 ontology.
- [3] Renato Iannella, Michael Steidl, Stuart Myles, and Víctor Rodríguez-Doncel. OdrI vocabulary & expression 2.2.
- [4] Renato Iannella and Serena Villata. OdrI information model 2.2.
- [5] Benedict Whittam Smith and Víctor Rodríguez-Doncel. OdrI implementation best practices.