Impact of Policies on Organizations Engaged in Partnership

Zakaria Maamar[1][a], Amel Benna[2][b] and Fadwa Yahya[3][c]
[1]*University of Doha for Science and Technology, Doha, Qatar*
[2]*Research Center for Scientific & Technical Information (CERIST), Algiers, Algeria*
[3]*Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia*

Keywords: Partnership, Policy, ODRL, Opposition, Support.

Abstract: This paper discusses how to identify and assess the impact of policies on organizations that engage in partnership. Partnership is a viable option for organizations wishing to sustain their growth and reinforce their competitiveness to respond to ongoing changing business conditions. Because policies regulate organizations' operations, they could either promote or undermine partnership. In this paper, policies are specialized into local and partnership, and are specified in Open Digital Rights Language (ODRL) defining what is permitted to do, what is prohibited from doing, and what must be done. The paper devises partnership scenarios as a set of coordinated partnership policies, labels partnership policies as either supportive of or opposing to local policies based on an impact analysis, and, finally, implements a policy-based partnership system.

1 INTRODUCTION

To sustain their growth and reinforce their competitiveness, organizations have multiple practical solutions to choose from such as revisiting their business practices, exploring new markets, and forming partnership that is the focus of this paper.

Despite the bright side of partnership because of benefits like bridging the gap in expertise and knowledge, securing more cash, and saving costs (Oelwang, 2022), the road ahead to a successful partnership could sometimes be "bumpy" minimizing any form of benefit and even, questioning the purpose of partnership. Organizations could be forced to adjust their structures and/or functions to accommodate each other's (sometimes conflicting) work cultures, business requirements, regulations, etc. It happens that an organization has Bring-Your-Own-Device (BYOD) policy that is not in-line with a partner's policy that restricts on-premise access to internal devices, only. What should organizations do in this case? Should they develop new policies which could result in expanding the partnership scope? Should they continue using existing policies which could result in contracting the partnership scope? Or, should they adjust existing policies which could, also, result in expanding the partnership scope with the risk of triggering conflicts with other policies? These are some questions that partnering organizations will have to address.

Aligning ourselves with how Rights Expression Languages (REL) like Open Digital Rights Language (ODRL) provide constructs (e.g., policies, constraints, and actions) to control (mainly digital) assets using permission, prohibition, and duty rules (W3C, 2018), we treat organizations' systems in this paper as assets and hence, will be the focus of partnership scenarios. The analysis of these scenarios would result in labeling partnership policies as either supportive or opposing depending on how they impact existing local policies of each partnering organization. This impact is a form of either solidification or worsening. On the one hand, a solidification example is when a group of employees can now access a partnering organization's systems off-premise; a duty rule in a partnership policy forces these employees to use a security protocol that outperforms what a partnering organization deploys. On the other hand, a worsening example is when another group of employees now see their access privileges reduced; a permission rule in a partnership policy refers to a system that is technically incompatible with a partnering organization's systems. Although the current literature focuses on conflicting policies as a potential "side effect" of partnership (Donghun, 2018; Liu et al., 2021;

[a] https://orcid.org/0000-0003-4462-8337
[b] https://orcid.org/0000-0002-9076-5001
[c] https://orcid.org/0000-0003-4661-1344

Maamar, Z., Benna, A. and Yahya, F.
Impact of Policies on Organizations Engaged in Partnership.
DOI: 10.5220/0012718500003687

In *Proceedings of the 19th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2024)*, pages 684-691
ISBN: 978-989-758-696-5; ISSN: 2184-4895

Wu and Liu, 2014), our examples illustrate how different we are in the sense of examining the impact of partnership policies on local policies. Therefore, as a pre-requisite to any successful partnership, we raise the following questions: *how* are partnership policies coordinated when devising partnership scenarios, and *why* are partnership policies labelled as either supportive or opposing with respect to solidifying or worsening local policies?

In this paper, our contributions are, but not limited to, (*i*) devising partnership scenarios as a set of coordinated partnership policies,(*ii*) specification of partnership and local policies in ODRL, (*iii*) labeling partnership policies as either supportive of or opposing to local policies, and (*iv*) illustration of how partnership policies would either solidify or worsen local policies based on this labeling. In the following, Section 2 briefly presents ODRL, some related works, and a case study. Section 3 analyzes the impact of policies on partnering organizations. Section 4 technically demonstrates this impact. Finally, Section 5 concludes the paper and identifies future work.

2 BACKGROUND

After an overview of ODRL and related works, the rest of this section discusses a case study.

2.1 ODRL in Brief

ODRL provides a flexible and interoperable information model, vocabulary, and encoding mechanisms to represent statements about the usage of assets. An asset is an identifiable resource or a collection of resources such as data/information, content/media, and services. ODRL adopts policies to represent permitted and prohibited actions over an asset and required obligations that stakeholders must meet over this asset as well (W3C, 2018). ODRL key constructs include:

- Policy could include permission, prohibition, or duty rules. First, permission allows an action over an asset if all constraints/duties are satisfied/fulfilled. Second, prohibition disallows an action over an asset if all constraints are satisfied. Finally, duty forcibly exercises an action over an asset or not. It is fulfilled when all constraints are satisfied and have been exercised.
- Party is an entity or a collection of entities that could be a person, group of persons, organization, or agent. A party can fulfill multiple roles namely, assigner, assignee, informed party, consented party, consenting party, compensated party, and tracked party.
- Constraint is used either to refine components like action, party, or collection of assets or to declare conditions applicable to a rule. Constraints could also be combined using logical operators.

Multiple works discuss ODRL from different perspectives and in different contexts. For instance, Kasten and Grimm use OWL to enrich ODRL with semantics (Kasten and Grimm, 2010). They note that ODRL is not meant for specific case studies and hence, can be adopted in many applications using the concept of license. As a result, ODRL is attractive to both policy advocates and ICT practitioners. In another work, Serrão et al. develop OpenSDRM system that uses ODRL as a REL in 3 different situations: digital music e-commerce, video-surveillance data streaming, and controlled access to remote sensing images (Serrão et al., 2005). In conjunction with ODRL, OpenSDRM adopts SOA's principles along with its SOAP, UDDI, and WSDL technologies.

2.2 Related Work

To the best of our knowledge, there are not works that address the issue of policies impacting other policies in term of either support or opposition. To address this limitation, we examine conflicting policies as a potential result of partnership.

In (Donghun, 2018), Yoon describes policy conflicts to manage research equipment as *"the conflict of interest between the government and the scientists in the public R&D business and its execution by the central government, local government, public institution, etc."*. Upon analysis of these conflicts' causes, 5 strategies were devised: formulate policies for research equipment relocation, hold negotiations for partial concession from one party to another, diffuse confrontation with respect to the issues causing conflicts, mitigate differences between conflicting parties, and promote rewards and mutual benefits. All these strategies address conflicts between the government and scientists to improve cooperation.

In (Khakpour et al., 2010), Khakpour et al. propose PobSAM that is an actor-based model for the development of self-adaptive systems. PobSAM uses policies to control and adapt systems' behaviors. In addition, PobSAM considers a classification of conflicts that could exist between a system's policies. These conflict types are expressed using a set of linear temporal logic patterns that are then, used to analyze policies and identify potential conflicts. Finally, PobSAM adopts static analysis and model checking technique to verify the correctness of system adaptation.

In (Liu et al., 2021), Liu et al. focus on attribute-based access control policy defined in eXtensible Access Control Markup Language to propose a formal definition of conflicts between a pair of rules in terms of explicit, probable, and never. A conflict occurs if there exists a user's request such as 2 rules having attributes with intersecting value sets that are applicable for this request, or the decisions of 2 rules like permit and deny are contradicting. A conflict is explicit if (1) one rule shares all attributes with another rule, (2) the shared attributes have intersecting value sets, (3) their action sets overlap, and (4) their decisions are distinct. A conflict is probable if the rules' action sets overlap and their decisions are distinct, and the attributes of rules generated from the main rules have intersecting value sets. Finally, there is no conflict when either the rules' decisions are the same, the rules' action sets do not overlap, or the rules' decisions are distinct and their action sets overlap. In either case, the rules are expected to have a subject, or an object attribute expression (with a particular attribute) that has non-overlapped value sets.

In (Wu and Liu, 2014), Wu and Liu address policy conflicts for collaborative service computing applications. Services are designed and deployed to work according to providers' policies. However, when service collaboration is required to accomplish specific business tasks, there could be some conflicts between their respective providers. 3 conflict types are considered: duty, interest, and outcome. To address them, the authors propose a knowledge-augmented logical analysis framework that includes temporal logic analysis rules, semantic extensions, reconciliation rules, and other domain information. First, a temporal first-order logic is used to express policies, so that the obtained logical expressions are analysed to deduce the existence/absence of conflicts between these policies. This analysis is augmented by a semantic knowledge base containing among others attribute relationships and dynamic constraints between attributes. The logical analysis framework uses this information in the logic reasoning to address the conflict.

Regarding ODRL, conflicts could occur between permission and prohibition rules included in the same policy, or when merging policies because of policy inheritance and the resultant rules are inconsistent. To address such conflicts, a policy with the conflict property as either perm, prohibit, or invalid could be used. Perm value means that a permission rule must override a prohibition rule. Prohibit value means that a prohibition rule must override a permission rule. Finally, invalid value applies when the entire policy is made void because of a conflict detected between rules after a policy merge or inheritance.

The works above address conflicting policies using different techniques. However, there is a limited emphasis on potential relations like support and opposition that would capture the impact of policies on other policies. In this paper, we use constraints to establish such relations along with using ODRL to specify these policies and their constraints.

2.3 Case Study

Our case study involves 2 organizations, *weRetail* that operates in on-line shopping and *weDeliver* that couriers parcels to customers' locations. *weRetail* runs online systems (assets according to ODRL vocabulary) that policies manage in terms of who can consult what, for what purpose, etc. Listing 1 is a policy between *weRetail* (line 6) and a customer (line 7). The customer sets 48 hours (lines 11-13) prior to the delivery date as a time limit (lines 15-17) for *weRetail* to update (line 9) her order like substituting items and changing delivery dates. Passed this time limit, the update is prohibited (line 4) unless *weRetail* (line 21) compensates the customer (line 22).

Listing 1: ODRL specification for customer orders.

```
1 {"@context": "http://www.w3.org/ns/odrl.jsonld",
2  "uid": "http://example.com/policy:304",
3  "@type": "Request",
4  "prohibition": [{
5      "target":"http://example.com/weRetail/
          asset/CustomerOrders",
6      "assigner":"http://example.com/weRetail",
7      "assignee":"http://example.com/users/
          customer",
8      "action": [{
9         "rdf:value": {"@id": "odrl:use"},
10        "refinement": [{
11           "leftOperand": "elapsedTime",
12           "operator": "lt",
13           "rightOperand": {"@value": "48H", "
                @type": "xsd:duration"},
14           "comment": "less than 48 h"},
15           {"leftOperand": "dateTime",
16            "operator": "gteq",
17            "rightOperand": "https://www.
                 wikidata.org/wiki/Q20058793",
18            "comment": "delivery date"}]}],
19     "remedy": [{
20         "action":"compensate",
21         "compensatingParty":"http://example.
              com/weRetail",
22         "compensatedParty":"http://example.com
              /users/customer"
23  }]}]}
```

The description above also applies to *weDeliver* that has policies like in Listing 2. The main differences with Listing 1 reside in the parcel scheduling system that is the asset (line 5), the collection of employees who are the assignee (lines 7-9), and the permission rule (line 4) that is constrained by a duty rule (line 11). This rule obliges *weDeliver*'s employees to request an access consent twice (lines 14-17) from a consenting party (line 18).

Listing 2: ODRL specification for parcel scheduling.

```
1 {"@context": "http://www.w3.org/ns/odrl.jsonld",
2  "uid": "http://example.com/policy:201",
3  "@type": "Agreement",
4  "permission": [{
5     "target": "http://example.com/weDeliver/
          asset/parcelSchedulingSystem",
6     "assigner":"http://example.com/weDeliver",
7     "assignee":{
8        "@type": "PartyCollection",
9        "source": "http://example.com/weDeliver/
             employees/"},
10    "action": "execute",
11    "duty":[{
12       "action":[{
13         "rdf:value": {"@id":"odrl:
               obtainConsent"},
14         "refinement": [{
15           "leftOperand": "count",
16           "operator": "eq",
17           "rightOperand": "2",
18           "unit": "https://www.wikidata.org/
                 wiki/Q11563"}]}],
19       "consentingParty": "http://example.com/
             parcelSchedulingSystem/consentment"}
             ]}]}
```

Handling joint customers' order and delivery transactions, *weRetail* and *weDeliver* combine their efforts during peak-seasons giving their respective employees the opportunity of temporarily accessing each other's systems while maintaining full control of these systems (Listings 1 and 2). For instance, *weRetail* will schedule, without *weDeliver* intervention, immediate deliveries to consumers who spend more than $2000 in a single transaction per day. And, *weDeliver* will modify orders' delivery dates without the intervention of *weRetail*, should some delivery time-slots turn out slightly loaded and hence, could be filled out with some early deliveries. These 2 scenarios shed light on some requirements that partnership policies should capture like the temporary nature of partnership, eligible employees from each organization to access each other's systems, etc. Below are 2 partnership policies exemplifying how local policies become solidified and worsened, respectively, based on how these partnership policies are defined.

1. Partnership policy solidifying Listing 2's local policy where a new assignee is linked to an existing asset. To allow John from *weRetail* to access *weDeliver*'s parcel scheduling system, he has to be in a specific location and to secure at least the same number of access consents from the same consenting party. In Listing 3, the partnership policy has John as an assignee who is given access to the parcel scheduling system as per the agreement between both organizations. The solidification of Listing 2's local policy with respect to Listing 3's partnership policy is due to both the access from a specific location and the adoption of at least the same number of consents.

Listing 3: ODRL specification for a partnership policy (1).

```
{"@context": "http://www.w3.org/ns/odrl.
     jsonld",
"uid": "http://example.com/policy:302",
"@type": "Agreement",
"permission": [{
   "target": "http://example.com/weDeliver/
        asset/parcelSchedulingSystem",
   "assigner":"http://example.com/weDeliver",
   "assignee":"http://example.com/weRetail/
        users/John",
   "action":[{
      "rdf:value": {"@id": "execute"},
      "refinement": [{
        "leftOperand": "spatial",
        "operator": "eq",
        "rightOperand":"https://www.wikidata.
             org/wiki/Q183?wprov=srpw1_0",
        "comment":"Germany"}]}],
    "duty":[{
       "action": [{
        "rdf:value": {"@id": "odrl:
             obtainConsent"},
        "refinement": [{
           "leftOperand": "count",
           "operator": "gteq",
           "rightOperand": "2",
           "unit":"https://www.wikidata.org/
                wiki/Q11563"}]}],
     "consentingParty":"http://example.com/
         parcelSchedulingSystem/consentment"
         }]}]}
```

2. Partnership policy worsening Listing 1's local policy where a new assignee is linked to an existing asset. Should Michael from *weDeliver* modify the delivery date of any customer's order stored in *weRetail*'s system, then *weDeliver* will compensate the customer if the modified delivery date happens after the 48 hours time-limit set in Listing 4. The worsening of Listing 1's local policy with respect to Listing 4's partnership policy is due to the additional financial compensation.

Listing 4: ODRL specification for a partnership policy (2).

```
{"@context": "http://www.w3.org/ns/odrl.
     jsonld",
"uid": "http://example.com/policy:305",
"@type": "Agreement",
"prohibition": [{
```

```
5   "target": "http://example.com/weRetail/
        asset/CustomerOrders",
6   "assigner":"http://example.com/weRetail",
7   "assignee":"http://example.com/weDeliver/
        Michael",
8   "action": [{
9      "rdf:value": {"@id": "odrl:modify"},
10     "refinement": [{
11       "leftOperand": "elapsedTime",
12       "operator": "lt",
13       "rightOperand": {"@value": "48H", "
            @type": "xsd:duration"},
14       "comment": "less than 48 h"},
15       {"leftOperand": "dateTime",
16        "operator": "gteq",
17        "rightOperand":"https://www.wikidata
            .org/wiki/Q20058793",
18        "comment": "delivery date"}]}],
19         ...,
20   "remedy": [{
21        "action":"compensate",...}]}]}
```

3 PARTNERSHIP POLICIES MANAGEMENT

This section manages policy-based partnership with a focus. labelling partnership policies as either supportive of or opposing to local policies.

3.1 Overview

Fig. 1 represents the chronology of operations that enables partnership between organizations using partnership policies and evaluate the impact of partnership policies on these organizations' local policies. It begins when organizations$_{i,j}$ develop local policies to manage their own assets (1) and then, engage in partnership like in Section 2.3. This engagement initiates developing joint scenarios. For each joint scenario, relevant bodies define partnership policies (2) and then, submit them to the *policy engine* for verification prior to their storage like the rest of local policies in a dedicated repository (3). Once the definition of all policies is complete, they are submitted to the *partnership engine* that coordinates partnership policies and next, labels them as either supportive of or opposing to local policies (4). More details about policy labeling are given in Section 3.2. Noticing both that supportive partnership policies solidify local policies and that opposing partnership policies worsen local policies, organizations$_{i,j}$ take measures to reinforce the solidification and offset the worsening, respectively (5).

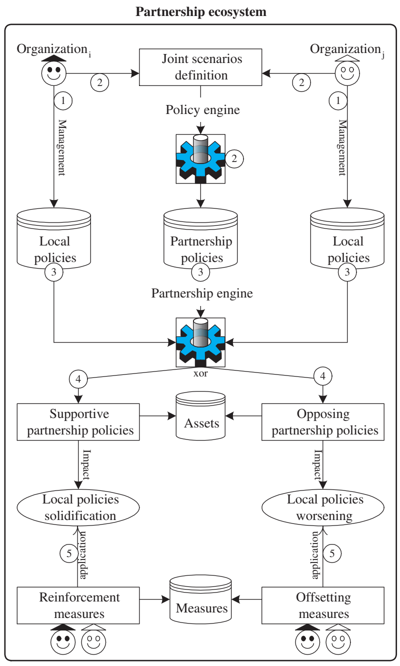

Figure 1: Policy-based management of partnership.

3.2 Labeling Partnership Policies

To label partnership policies as either supportive of or opposing to local policies, we look into potential overlaps between their respective constraints. In ODRL, the constraint construct applies to specific constructs namely, action, assignee, rule, and asset although constraints are optional in policies. To label partnership policies, introduce the prohibition-nullification concept and then, adopt ODRL constraint-structure to label a partnership policy. Fig. 2 summarizes the chronology of operations to adopt and the conditions to check underpinning this labeling.

Prohibition nullification converts a prohibition rule into a permission rule to ensure an "homogeneous" analysis of rules across all partnership and local policies. We select permission rule because of the lack of obligation to trigger a prohibition rule after conversion. Should a prohibition rule have constraints on itself, then their complementary constraints will be integrated into the permission rule. In addition, should a prohibition rule include remedies that compensate this rule's violation, then the remedies will become duty rules in the permission rule.

In ODRL, a constraint structure consists of 2 operands connected by 1 relational operator like in Listing 1's lines 10-12 where the *leftOperand*[1] is "elapsedTime", the *rightOperand* has a value of

[1] A predefined list of ODRL terms exists.

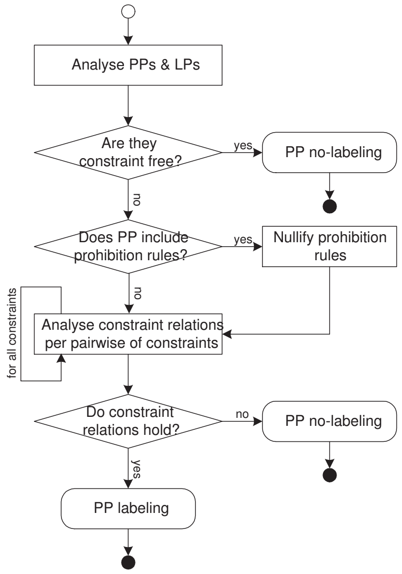

Figure 2: Flowchart of partnership policies labelling.

48 hours, and the *operator* is "lt". Thanks to the *leftOperand*, we categorize constraint into budget, security, time, location, and personnel.

In conjunction with prohibition nullification, we define constraint relation between $\mathcal{P}$artnership $\mathcal{P}$olicies ($\mathcal{PP}$) and $\mathcal{L}$ocal $\mathcal{P}$olicies ($\mathcal{LP}$) using their respective constraints, $\mathcal{PP}_{i,C_{ik}}$ and $\mathcal{LP}_{j,C_{jl}}$ that apply to the same actions in these policies' rules. This relation allows to decide on the constraint that a partnership policy would use, i.e., $\mathcal{PP}_{i,C_{ik}}$, $\mathcal{PP}_{j,C_{jl}}$, either $\mathcal{PP}_{i,C_{ik}}$ or $\mathcal{PP}_{j,C_{jl}}$, or neither $\mathcal{PP}_{i,C_{ik}}$ nor $\mathcal{LP}_{j,C_{jl}}$, leading to label a $\mathcal{PP}$ as either supportive of, opposing to, partially supportive of/opposing to, or neutral to a $\mathcal{LP}$, respectively. The constraint relations are:

- consolidate($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$): when initiating $\mathcal{PP}_i$, $\mathcal{LP}_{j,C_{jl}}$ takes effect over $\mathcal{PP}_{i,C_{ik}}$. Should the consolidate relation hold, then $\mathcal{PP}_i$ will be labeled as supportive of $\mathcal{LP}_j$ for that particular constraint and hence, $\mathcal{LP}_{j,C_{jl}}$ will be adopted.
- override($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$): when initiating $\mathcal{PP}_i$, $\mathcal{PP}_{i,C_{ik}}$ takes effect over $\mathcal{LP}_{j,C_{jl}}$. Should the override relation hold, then $\mathcal{PP}_i$ will be labeled as opposing to $\mathcal{LP}_j$ for that particular constraint and hence, $\mathcal{PP}_{i,C_{ik}}$ would be adopted.
- nullify($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$): when initiating $\mathcal{PP}_i$, neither $\mathcal{PP}_{i,C_{ik}}$ nor $\mathcal{LP}_{j,C_{jl}}$ takes effect calling for adjusting $\mathcal{PP}_{i,C_{ik}}$ into $\mathcal{PP}_{i,C'_{ik}}$ without violating $\mathcal{LP}_{j,C_{jl}}$. Should the nullify relation hold, then $\mathcal{PP}_i$ will be labeled as partially supportive of/opposing to $\mathcal{LP}_j$ for that particular constraint and hence, $\mathcal{PP}_{i,C'_{ik}}$ will be adopted.
- relax($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$): when initiating $\mathcal{PP}_i$, either $\mathcal{PP}_{i,C_{ik}}$ or $\mathcal{LP}_{j,C_{jl}}$ takes effect. Should the relax relation hold, then $\mathcal{PP}_i$ will be labeled as neutral to $\mathcal{LP}_j$ for that particular constraint and hence, either $\mathcal{PP}_{i,C_{ik}}$ or $\mathcal{LP}_{j,C_{jl}}$ will be adopted.

Let us illustrate the discussions above with 3 examples including the case study. In the first example, $\mathcal{PP}_i$'s and $\mathcal{LP}_j$'s constraints are on the action construct. We assume that "dateTime" *leftOperand* in these constraints is defined as a time interval ($[b_{i|j}, e_{i|j}]$). To reason over time intervals, we resort to time relations (Allen, 1983):

1. Should $equals([b_i,e_i],[b_j,e_j])$ hold, then relax($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$) will exist indicating that $\mathcal{PP}_i$ is time-based neutral to $\mathcal{LP}_j$. $\mathcal{PP}_{i,C_{ik}}$'s time interval fully covers $\mathcal{LP}_{j,C_{jl}}$'s time interval and *vice versa*.
2. Should $during|starts|finishes([b_i,e_i],[b_j,e_j])$ hold, then override($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$) will exist indicating that $\mathcal{PP}_i$ is time-based opposing to $\mathcal{LP}_j$. $\mathcal{PP}_{i,C_{ik}}$'s time interval partially covers $\mathcal{LP}_{j,C_{jl}}$'s time interval.
3. Should $contains([b_i,e_i],[b_j,e_j])$ hold, then consolidate($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$) will exist indicating that $\mathcal{PP}_i$ is time-base supportive of $\mathcal{LP}_j$. $\mathcal{LP}_{j,C_{jl}}$'s time interval is totally covered by $\mathcal{PP}_{i,C_{ik}}$'s time interval.
4. Should $precedes|meets([b_i,e_i],[b_j,e_j])$ hold, then $\mathcal{PP}_i$ will not be labelled with regard to $\mathcal{LP}_j$.
5. Should $overlaps([b_i,e_i],[b_j,e_j]$ hold, then nullify ($\mathcal{PP}_{i,C_{ik}}$,$\mathcal{LP}_{j,C_{jl}}$) will exist indicating that $\mathcal{PP}_i$ is partially time-based supportive of/opposing to $\mathcal{LP}_j$. Some parts of $\mathcal{PP}_{i,C_{ik}}$'s time interval do not cover $\mathcal{LP}_{j,C_{jl}}$'s time interval leading to adjusting $\mathcal{PP}_{i,C_{ik}}$'s initial time interval to $[b_j,e_i]$.

Another example is the case study where $\mathcal{PP}$'s constraints in Listing 3 and $\mathcal{LP}$'s constraints in Listing 2 are on the action construct. To start with, the *execute* action is constraint free in the local policy, so this policy is dropped from the labeling of the partnership policy. Let us now focus on the *obtainConsent* action that is present in both policies. Based on the respective constraints on this action, consolidate($\mathcal{PP}_{C_{count}}$,$\mathcal{LP}_{C_{count}}$) exists because the partnership policy requires at least the same number of consents as the local policy does indicating that $\mathcal{PP}$ is security-based supportive of $\mathcal{LP}$.

```
Local Policy : target=http://example.com/weDeliver/asset/parcelSchedulingSystem,
 action=odrl:obtainConsent,
 leftOperand=count,
 operator=eq,
 rightOperand=2
Supportive
Partnership Policy : target=http://example.com/weDeliver/asset/parcelSchedulingSystem,
 action=odrl:obtainConsent,
 leftOperand=count,
 operator=gteq,
 rightOperand=2
```

Figure 3: Labeling of $\mathcal{PP}_1$ as supportive of $\mathcal{LP}_1$.

```
Local Policy : target=http://example.com/weDeliver/asset/parcelSchedulingSystem,
 action=odrl:execute,
 leftOperand=spatial,
 operator=eq,
 rightOperand=Japan
Opposing
Partnership Policy : target=http://example.com/weDeliver/asset/parcelSchedulingSystem,
 action=odrl:execute,
 leftOperand=spatial,
 operator=eq,
 rightOperand=Tokyo
```

Figure 4: Labeling of $\mathcal{PP}_2$ as opposing to $\mathcal{LP}_2$.

```
Local Policy : target=http://example.com/weDeliver/asset/parcelSchedulingSystem,
 action=odrl:execute,
 leftOperand=dateTime,
 operator=gt,
 rightOperand=2023-12-01
 operator=lteq,
 rightOperand=2024-01-31
Partially supportive of/opposing to
Partnership Policy : target=http://example.com/weDeliver/asset/parcelSchedulingSystem,
 action=odrl:execute,
 leftOperand=dateTime,
 operator=gt,
 rightOperand=2023-11-15
 operator=lt,
 rightOperand=2024-01-15
```

Figure 5: Labeling of $\mathcal{PP}_3$ as partially supportive of/opposing to $\mathcal{LP}_3$.

4 IMPLEMENTATION

To assess the impact of policies on partnering organizations' assets, a system was developed in Java 8 using Eclipse IDE for Java Developers and deployed on Windows 8 64 bits, 1.4 GHz quad-core processor CPU, 4GB RAM desktop. It mainly targets Fig. 1's *partnership engine* along with categorizing constraints (e.g., budget, time, and location), so that relevant policy managers in the system are enabled. The source code is released on GitHub at https://tinyurl.com/4sc367vn.

The system parses $\mathcal{LP}$s and $\mathcal{PP}$s to identify those pairwises of $\mathcal{LP}_i$ and $\mathcal{PP}_j$ that have similar assets and similar actions on these assets, so they could engage in supporting and/or opposing relations. The parsing is taken care of by an in-house JSON parser developed using json-simple Java library. Then, the system categorises these pairwises as per their respective constraints. For each categorized $\mathcal{LP}_i$ and $\mathcal{PP}_j$ pairwise, the system selects the adequate policy manager to label these policies using constraints' *leftOperand*. Depending on constraint categories, different tools are used such as Joda-Time java[2] library to deal with time and time-interval constraints, and

[2]https://www.joda.org/joda-time/

World Cities Database[3] to deal with location constraint. This database includes more than 4 million cities and towns in the world and can be queried to define relations like *equals*, *contains*, and *disjoint* between 2 locations.

To test the system, we adopted $\mathcal{LP}_1$ and $\mathcal{PP}_1$ of Listings 2 and 3, respectively, and defined more for instance, $\mathcal{LP}_{2,3}$ and $\mathcal{PP}_{2,3}$ integrating *spatial* and *dateTime* ODRL constraints. An overview of these policies is as follows: $\mathcal{PP}_2$ (the use of the parcel scheduling system as an asset is constrained to a specific location that is Tokyo), $\mathcal{LP}_2$ (the use of the parcel scheduling system as an asset is constrained to a specific location that is Japan), $\mathcal{PP}_3$ (the use of the parcel scheduling system as an asset is constrained to a specific time-period that is 2023-11-15 to 2024-01-15), and $\mathcal{LP}_3$ (the use of the parcel scheduling system as an asset is constrained to a specific time-period that is 2023-12-01 to 2024-01-31).

Afterwards, we submitted the policies to the system for labeling. Fig. 3 shows that $\mathcal{PP}_1$ is supportive of $\mathcal{LP}_1$ due to consolidate relation. Fig. 4 shows that $\mathcal{PP}_2$ is opposing to $\mathcal{LP}_2$ because of override relation. Finally, Fig. 5 shows that $\mathcal{PP}_3$ is partially supportive of/opposing to $\mathcal{LP}_3$ because of nullify relation.

5 CONCLUSION

This paper discussed partnership as a viable option for organizations wishing to bridge the gap in expertise and knowledge, to secure more cash, and to save costs. However, partnership could run into obstacles minimizing its benefits and even, questioning its purpose when organizations end-up adjusting their structures and/or functions to accommodate each other's (sometimes conflicting) work cultures, business requirements, and regulations. These regulations correspond to policies that we specify in ODRL and specialize into partnership and local. To examine potential relationship between these 2 types of policies, we devised partnership scenarios as a set of policies to coordinate, labeled partnership policies as either supportive of or opposing to local policies, and finally demonstrated a system implementing policy-based partnership targeting a case study involving weRetail that operates in the domain of on-line shopping and weDeliver that operates in the domain of couriering parcels to customers' locations. In term of future work, we would like to define measures that either reinforce the solidification of or offset the worsening of local policies, assess the scalability of the system when several policies need to be managed, and implement the reinforcement and offsetting measures.

[3]simplemaps.com/data/world-cities.

REFERENCES

Allen, J. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11).

Donghun, Y. (2018). The Policy Conflict Research of Interested Parties for the Efficient Management of Research Equipment: With Focus on the Government and the Scientist. *Cogent Business & Management*, 5(1).

Kasten, A. and Grimm, R. (2010). Making the Semantics of ODRL and URM Explicit Using Web Ontologies. In *Proceedings of Goods'2010*, Namur, Belgium.

Khakpour, N., Khosravi, R., Sirjani, M., and Jalili, S. (2010). Formal Analysis of Policy-based Self-Adaptive Systems. In *Proceedings of SAC'2010*, Sierre, Switzerland.

Liu, G., Pei, W., Tian, Y., Liu, C., and Li, S. (2021). A Novel Conflict Detection Method for ABAC Security Policies. *Journal of Industrial Information Integration*, 22.

Oelwang, J. (2022). *Partnering*. Optimism Press.

Serrão, C., Dias, M., and Delgado, J. (2005). Using ODRL to Express Rights for Different Content Usage Scenarios. Technical report, Iscte Repository, University Institute of Lisbon, Portugal, https://repositorio.iscte-iul.pt/handle/10071/250.

W3C (2018). ODRL Information Model 2.2. https://www.w3.org/TR/2018/REC-odrl-model-20180215/. (Visited January 2022).

Wu, Z. and Liu, Y. (2014). Knowledge Augmented Policy Conflict Analysis for Services Collaboration. *Knowledge-Based Systems*, 62.