Semantic-Driven Enforcement of Rights Delegation Policies via the Combination of Rules and Ontologies [*]

Yuh-Jong Hu

Emerging Network Technology (ENT) Lab.
Dept. of Computer Science
National Chengchi University,
Taipei, Taiwan, 11605,
hu@cs.nccu.edu.tw

Abstract. We show that the semantic formal model for Open Digital Right Language (ODRL)-based rights delegation policies can be enforced and expressed as a combination of ontologies and rules, e.g., Semantic Web Rule Language (SWRL). Based on ODRL's expressions and data dictionary, a rights delegation ontology is proposed in this study. Furthermore, we express the rights delegation policy as a set of ontology statements, rules, and facts for usage and transfer rights delegation. When verifying ODRL formal semantics, our SWRL approach is superior to the generic restricted First Order Logic (FOL) model because we have an understandable formal semantics of policies for automatic machine processing and a higher expressive power for policy compliance checking. On the other hand, the rights delegation semantics shown as a generic full FOL might have a higher complexity of license verification, which results in a policy compliance checking that is possibly undecidable. A real usage rights delegation scenario for digital content is demonstrated in order to justify the feasibility of our formal semantic model for digital rights delegation. We hope this study will shed some light on future sensitive information usage and delegation rights controlled from a privacy protection perspective.

1 Introduction

The ultimate goal of achieving a distributed Digital Rights Management (DRM) system is content owners can project policies governing their content into remote environments with confidence that those policies will be respected by remote nodes [13]. A node is a trusted system that governs the legal usage of digital works that can be relied on to follow certain rules and enforce its legal rights delegation policy [19]. Aspects of the DRM rights authorization and enforcement problem include formulating delegation policies and a mechanism for "proving"

[*]This research was partially supported by the Taiwan National Science Council (NSC), under Grant No. NSC 95-2221-E-004-001-MY3.

that a request to access rights complies with relevant policies. A general-purpose Rights Expression Language (REL) is a type of policy delegation language where the focus of the language is on the expression and transference of usage rights or capabilities from one party to another in an interoperable manner. It will be a challenge to design a general-purpose REL for the DRM system that expresses rights delegation policies and controls digital content [13]. Emerging acceptable industry REL are classified into two major camps: Open Digital Right Language (ODRL) and eXtensible rights Markup Language (XrML). Unfortunately, the semantics of both of these RELs are either described in English or as computer algorithms, therefore, they lack machine understandable formal semantics.

There are two core components for a DRM rights delegation policy: an REL language for expressing policies and an evaluator that can make decisions based on such expressions. The policy evaluator must be able to reason correctly concerning all types of policy it may encounter when making a trusted decision to grant rights. Thus, the design of a policy evaluator is going to be influenced by the design of the REL language. A DRM policy evaluator must decide for each requested access whether the policy (or policies) is relevant to the request and whether or not to allow it to occur for a given license. This formulation of a DRM policy evaluation can be regarded as a "compliance checking" decision problem in a trust management system [1]. The license is derived from a legal contract that states the permissible agreements under which digital contents can be legitimately accessed. The languages for writing licenses (or permissible agreements) usually fall into three categories: a human readable natural language, a software readable XML-based language, and a machine understandable language [18].

When we consider digital contents as protected, sensitive, personal information which might be disseminated over the entire Web, then the usage and delegation rights control issues we are facing are just the same as those existing in the DRM system. Disseminated digital content (or information) with associated licenses are encrypted with appropriate security keys. If a node with a service request can decrypt the downloaded information and license, then the node's embedded license evaluator will faithfully interpret the license semantics and enforce the license agreements, including ontology, rules, and facts, to decide whether a request should be granted or not.

2 Research Goal

The goal of this research is to deal with the problem that license agreements written in ODRL REL, are open to interpretation that results in semantic ambiguity. This is because the stated conditions for which resources access legitimate license are written in English. We need an abstract semantic layer that can be overlaid on existing ODRL data models to express their license and service semantics instead of using natural language, such as English.

ODRL is one of the most popular RELs for expressing digital license exchange and sharing, it also has an XML-based markup language. As we know, XML has the capacity of marking up licenses and data for machine processing but does not have the capability of encoding the license semantics. The generic ODRL foundation model consists of three core entities: assets, rights, and parties. We are going to exploit this model by finding out which parts of license semantics can be shown as ontology language and which parts can be shown as rule language.

Therefore, DRM ontology and rights delegation policies will be using machines to ensure their license semantics. Finally, we show that our flexible rights delegation model could explicitly declare and enforce all kinds of rights delegation semantics through existing ODRL expressions and data dictionaries.

2.1 Our Approach

The formal semantics we propose are based on Semantic Web Rule Language (SWRL) [9]. SWRL is a language that combines description logic OWL with logic program rule language, such as RuleML Lite (see http://www.ruleml.org/#Lite.), where a Horn clause rules with the extension to OWL that overcomes many limitations of property chaining [9]. Property chaining features allow us to "transfer rights" from one class of individuals to another via delegation properties other than subClassOf rights inheritance.

In ODRL, possible permission usage rights are display, print, play, and execute. Possible permission transfer rights are usually defined as rights for rights, including sell, lend, give, and lease, etc [10]. Property chaining is a necessary feature for allowing rights delegation policies to delegate rights from one party to another when they belong to different classes. This important feature is not supported by other ontology-based semantic web policy languages, such as KAoS, Rei [20]. However, there are some limitations when using SWRL due to predicates being limited to being OWL classes and properties that only have a maximum parity of two, with no built-in arithmetic predicates or nonmonotonic features [5][9]. Therefore, we use OWL's extended concrete datatypes with unary and binary arithmetic operators in license agreement verification so that the verifier can verify whether prerequisite requirements and constraints in a license are compliant with its rights delegation policy [16].

When verifying ODRL formal semantics, the ontology+rule (SWRL) approach is superior to the generic restricted First Order Logic (FOL) formal semantics model [18]. First, generic restricted FOL-based rights delegation policies cannot be automatically processed by an agent because these FOL-based policies lack a semantic rights markup language. Second, unlike our SWRL (Ontologies+Rules) policies, restricted generic FOL policies do not have a high level of expressive power in their delegation policies [8]. On the other hand, the rights delegation semantics shown as a generic full FOL model might have a higher complexity of license verification, which results in a policy compliance checking that is possibly undecidable. Finally, Descritpion Logic (DL) in SWRL is possibly augmented by unary and binary arithmetic operators to enhance its concrete datatype operation [2].

3 Related Work

DRM and other modern access controls, such as privacy protection, RBAC, etc, are all regarded as $UCON_{ABC}$ models which integrate *Authorization (A), oBligations (B), and Conditions (C)* elements. Usage control is a generalization of access control that covers authorization, obligation, conditions, continuity (ongoing controls), and mutability [17]. In [21], a rule-based policy management system can be deployed in an open and distributed WWW site by creating a "policy aware" infrastructure. This makes the widespread deployment of rules and proofs on the Web to become a reality. However, this server-based access control infrastructure cannot be applied to DRM or other methods of privacy protection for usage and rights delegation control where information might be disseminated over the entire Web. Delegation Logic, a datalog-extended tractable logic-based language with expression of delegation depth and complex principals was proposed to represent policies, credentials, and requests in distributed authorization. However, it did not have a rights markup language to explicitly encode rights delegation ontology for automatic agent processing of its rights delegation semantics [14].

XrML does not have formal semantics [3]. Instead, the XrML specification presents semantics in two ways: as an English description of the language or as an algorithm that determines if rights are permissible from a set of licenses. A formal foundation model for XrML semantics is shown as FOL-based rights expression statements [7]. ODRL is another popular XML-based REL language used to state the conditions under which resources can be legitimately accessed [10]. ODRL does not have formal semantics either. The meaning of the language's syntax is described in English; license agreements written in ODRL are open to interpretation that results in semantic ambiguity. In order to resolve this problem, a formal foundation model for ODRL semantics is shown as a generic restricted FOL but it has less expressive power on rights expression and delegation as our SWRL approach [18]. In [6], they only provide a generic representation of contract information on top of RELs so that the enforcement of access rights can be extracted from ODRL-based digital license contracts. But, machine understandable formal semantics cannot be represented and processed in this study. In [4], an OWL-based ODRL formal semantic model is designed and deployed but it does not have usage and transfer rights delegation service capability. In summary, a formal foundation for ODRL or XrML semantics are shown as either FOL or OWL, but they all lack semantic-driven enforcement of rights delegation policies [4][18].

4 License Agreement for Usage Rights

The central construct of ODRL is a license agreement. A license agreement indicates the policies (rules) under which a principal $Prin_o$ allows another principal $Prin_{u_i}$ to use an asset r presumably owned by $Prin_o$, where $Prin_o$ is an asset owner and $Prin_{u_i}$ is one of n asset users, where $i \in (1, \cdots, n)$. A license agreement refers to a policy set showing any number of prerequisites and policies. A prerequisite is either a constraint, a requirement, or a condition. Constraints are facts that are outside the $Prin_{u_i}$'s influence but are defined by the asset owner $Prin_o$, such as counting or temporal restrictions for digital asset usage rights. Requirements are facts that are within the $Prin_{u_i}$ user's power to meet, such as prepaid fees before using a particular asset. Conditions are constraints that must *not* hold exceptions [18]. If all of the prerequisites are met, then a policy says that the agreement's users may perform the action for the license agreement's assets.

4.1 Rights Delegation Ontology

ODRL does not enforce or mandate any policies for DRM, but provides mechanisms to express such policies. ODRL specifications contain expression language, data dictionary elements, and XML syntax to encode the ODRL expressions and elements [10]. We are going to use these ODRL expression language and data dictionary elements as our rights delegation ontology's entities (see Fig. 1). The source of this ontology conceptualization is based on the ODRL 1.1 specification explicitly defining the ODRL's rights delegation semantics for a license in this ontology [10]. The class and property terms defined in this rights delegation ontology will be considered as antecedents or conclusion(s) in the following usage rights delegation policies to enforce all kinds of real rights delegation inference (see Section 5.2).

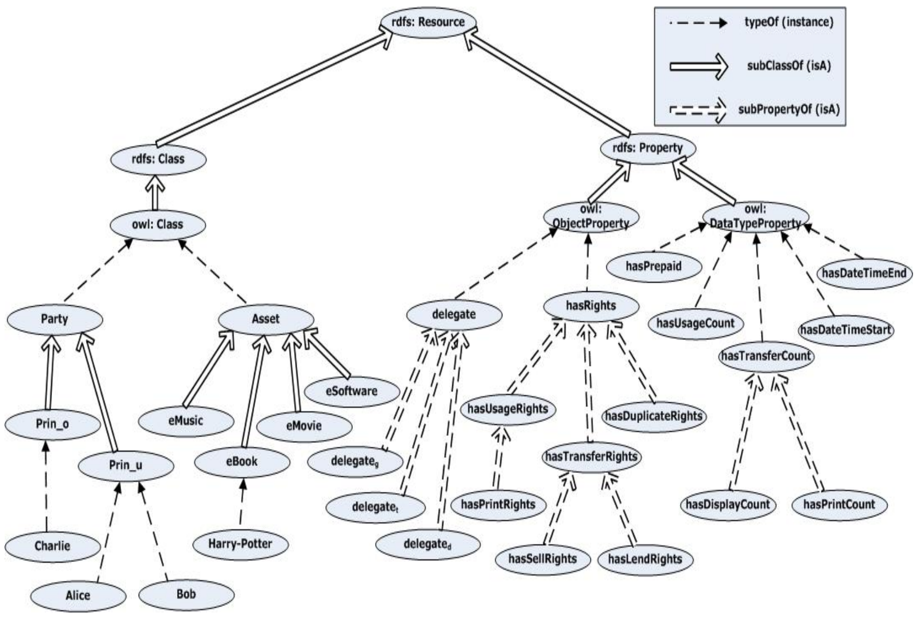

Fig. 1. A rights delegation ontology for an ODRL foundation model based on [10]

4.2 Usage Rights Delegation

We define $hasUsageRights$ as an abstract property describing the generic usage rights for a principal x to use an asset r. The domain class of $hasUsageRights$ property is $Party$, and the range class is Asset (see Fig. 2). The domain class of $delegate$ property is $Prin_o$ and the range class is $Prin_u$, where the delegate does have $subPropertyOf$ $(delegate_g, delegate_t, \cdots)$. The $delegate_g$ represents generic usage rights delegation property and the $delegate_t$ represents rights transfer delegation property. We do not allow a principal x be able to delegate his or her generic rights to another principal y if that principal x only has some usage rights but does not have any permissible transfer rights.

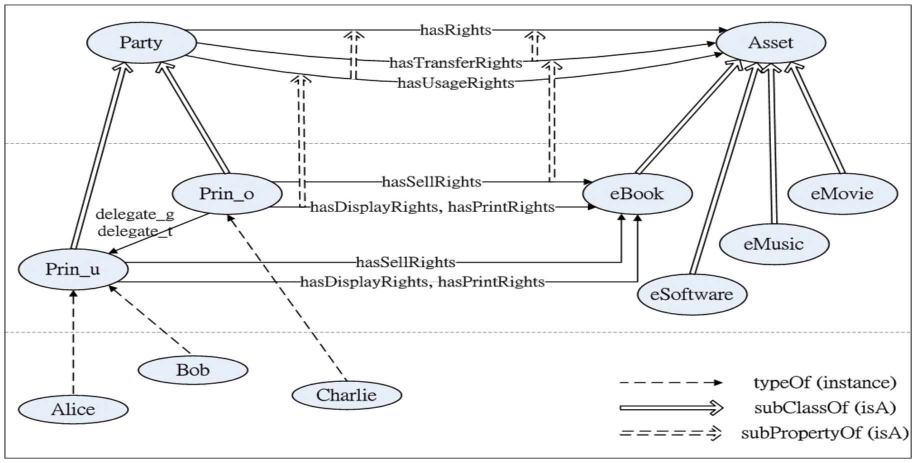

Fig. 2. A rights delegation snapshot based on rights delegation ontology

5 License Agreement for Transfer Rights

The delegation processes for transfer rights license agreements are activated using $delegate_t$ property, where the rights receiver owns the delegated rights but the rights owner might lose the rights temporarily or permanently. This is not true for some rights delegation scenarios where the rights owner and the rights receiver might have their rights concurrently. Thus, we create a rights **duplicate** delegation, indicated as $delegate_d$ from a variety of transfer rights. In this rights duplicate delegation property $delegate_d$, the rights original owner, concurrently has his or her own rights with the rights receiver after the rights duplicate delegation process is completed.

5.1 Prerequisites Expressions

We found that downstream rights receivers are receiving less rights in the rights delegation chain. An original content creator usually specifies his or her usage and transfer rights delegation with a reasonable number of depth d by constraint of $\leq_{\exists d}$ $hasTransferCount$, where d is a constant and is decreased by one for each delegation. Thus, the rights delegation process can be enabled as long as the condition $\geq_1$ $hasTransferCount$ is truth in a delegation policy (see rule ($o4$) in Section 5.2). In summary, we use extended OWL's unary arithmetic operators to express a prerequisite that can be a constraint, a requirement, a condition, or even a delegation depth.

Constraints for prerequisite such as, prepaid conditions, permissible count of upper (or exact) limit of usage rights, permissible delegation depth of transfer rights, and the validity time interval of usage rights, are shown in the following:

- MaxCardinality:$\leq_{\exists u}$ $hasUsageCount_{\exists p}.Asset$
- MaxCardinality: $\leq_{\exists t}$ $hasTransferCount_{\exists p}.Asset$
- Cardinality: $=_{\exists a}$ $hasPrepaid_{\exists p}.Party$
- Validity of time interval $\forall Time \in (t_1, t_2)$:
  $\geq_{\exists t_1}$ $hasDateTime_{\exists p}.Time \wedge \exists \leq_{t_2}$ $hasDateTime_{\exists p}.Time$

Sometimes, the usage rights prerequisite is enforced by a principal who is in charge of a counting action that collects all necessary mutable facts from the downstream rights receivers in the delegation chain. We show conditions as the following, where $\exists =_{\exists u}$ $hasUsageCount_{\exists p}.Asset$, $\exists \geq_{\exists t_1}$ $hasDateTime_{\exists p}.Time$, $\exists \leq_{\exists t_2}$ $hasDateTime_{\exists p}.Time$, and $\exists hasPrepaid_{\exists p}.Party$, indicate which principal p is in charge of mutable constraint parameter computations and policy compliance checking [15]. All of these will be demonstrated in Section 5.3.

5.2 Usage Rights Transfer Delegation

The $hasTransferRights$ is an abstract property describing the transfer rights delegation of usage rights for a principal x for an asset r. The domain class of property $hasTransferRights$ is $Party$ and the range class is Asset. $Prin_o$ might use $delegate_g$ to transfer usage rights only to $Prin_{u_i}$, where $i \in (1, \cdots, n)$, but does not delegate his transfer rights to $Prin_{u_i}$, where transfer rights $\in$ ($hasSell_tRights, \cdots$). Therefore, each $Prin_{u_i}$ cannot further delegate his usage rights to another $Prin_{u_j}$ (see rule ($o2$)). If we use $delegate_t$ property, then any one of the transfer rights permissions $\in$ ($hasSell_tRights, \cdots$) and usage rights can be further propagated (see rule ($o4$)). The depth of transfer delegation can be specified in class Asset with cardinality shown as $\exists =_{\exists n}$ $hasTransferCount.eBook$, which indicates that the transfer rights permission for $eBook$ can be propagated with the exact delegation depth of n:

- If party x has both usage and transfer rights for asset r, then he or she is allowed to transfer full (or partial) of both rights to another party but he or she can not keep his or her own rights after delegation:

$hasUsageRights(?x, ?r) \land hasTransferRights(?x, ?r)$
$\Longrightarrow hasUsageTransferRights(?x, ?r) \longleftarrow (o1)$

– Party x can only transfer his or her usage rights for asset r to another party y if his or her cumulative depleted usage count $<_{\exists u}$, where u is a constant indicating a count of upper limit of usage rights. Here, party y can have delegated usage rights but cannot have further delegation rights:

  $hasUsageTransferRights(?x, ?r) \land delegate_g(?x, ?y) \land hasPrepaid(?y, ?a) \land$
  $<_{\exists u} hasUsageCount(?r) \Longrightarrow hasUsageRights(?y, ?r) \longleftarrow (o2)$

– If party x has usage rights permission for resource r and the cumulative depleted usage count is $<_{\exists u}$. Furthermore, party x's current local date and time $t \in (t_1, t_2)$, then he or she is permitted to have these particular usage action, such as play, display, or print, etc:

  $hasUsageRights(?x, ?r) \land <_{\exists u} hasUsageCount(?r) \land \geq_{\exists t_1} hasDateTime(?t)$
  $\land \leq_{\exists t_2} hasDateTime(?t) \Longrightarrow Permitted(Usage, ?r) \longleftarrow (o3)$

– Party x can transfer his or her usage and transfer rights for asset r to another party y so party y can have x's both rights to transfer rights forward as long as x is not the final node in a delegation chain:

  $hasUsageTransferRights(?x, ?r) \land delegate_t(?x, ?y) \land hasPrepaid(?y, ?a) \land$
  $\geq_1 hasTransferCount(?r) \Longrightarrow hasUsageTransferRights(?y, ?r) \longleftarrow (o4)$

5.3 A Usage Rights Delegation Scenario

The following license agreement for a usage rights delegation scenario is adopted and modified from [18]. For reasons of space, a detailed discussion of the implications of our complete operational semantics for this scenario is left to the full paper for further study. This might need a speech-act agent communication language to represent message passing ontology, which would then allow our agents to automatically exchange interactive information among themselves as shown in [11]:

– **Natural Language (NL)** denotation of license agreement:

  Content distributor *Charlie c* makes an agreement with two content consumers, *Alice a* and *Bob b*. After each paying five dollars, and then both receiving acknowledgement from *Charlie*, *Alice* and *Bob* are given the usage rights and may each display an *eBook* asset, *Harry Potter and the Deathly Hallows*, up to five times. They may each print it only once. However, the total number of actions, either displays or prints done by *Alice* and *Bob*, may be at most ten. The usage rights validity period is between 2007/05/07/09:00 - 2007/05/10/24:00.

– **Human Readable Abstract Syntax** denotation of license agreement:

```
agreement
between Charlie and {Alice,Bob}
about Harry Potter and the Deathly Hallows
with inSequence[prePay[5.00],attribution[Charlie]]
|==> not[and[Time < 2007/05/07/09:00,Time > 2007/05/10/24:00]]
|==> with count[10] ==>
and[forEachMember[{Alice,Bob};count[5]] ==> display,
     forEachMember[{Alice,Bob};count[1]] ==> print]
```

– **First Order Logic (FOL)** denotation of license agreement:

$$\begin{aligned}&\forall x((x = \mathit{Alice} \lor x = \mathit{Bob}) \Longrightarrow\\&\exists t_1 \exists t_2 (t_1 < t_2 \land \mathit{Paid}(5, t_1) \land \mathit{Attributed}(\mathit{Charlie}, t_2))) \Longrightarrow\\&\forall t \land \mathit{hasDateTime}(t) \geq 2007/05/07/09:00 \land\\&\mathit{hasDateTime}(t) \leq 2007/05/10/24:00 \Longrightarrow\\&\mathit{count}(\mathit{Alice}, \mathit{id}_1) + \mathit{count}(\mathit{Alice}, \mathit{id}_2) + \mathit{count}(\mathit{Bob}, \mathit{id}_1)\\&+ \mathit{count}(\mathit{Bob}, \mathit{id}_2) < 10 \Longrightarrow\\&(\mathit{count}(\mathit{Alice}, \mathit{id}_1) < 5 \land \mathit{count}(\mathit{Bob}, \mathit{id}_1) < 5 \Longrightarrow \mathbf{Permitted}(x, \mathit{display}, \mathit{ebook}))\\&\land (\mathit{count}(\mathit{Alice}, \mathit{id}_2) < 1 \land \mathit{count}(\mathit{Bob}, \mathit{id}_2) < 1 \Longrightarrow \mathbf{Permitted}(x, \mathit{print}, \mathit{ebook})))\end{aligned}$$

We use the ontologies+rules (SWRL) approach to enforce the semantics of rights delegation policies instead of the above pure FOL-based formula. The following ontology, rules, and facts are a partial view from distributor *Charlie c* based on Fig. 1 and Fig. 2. In the bootstrapping stage, *Charlie c* has all of the usage and transfer (or duplicate) rights for the *eBook* class, including *HarryPotter and the Deathly Hallows*, which are shown as the facts in the following page. Ontology statements ($c1$) - ($c3$) indicate the constraints of associated usage counts shown in the above FOL formula. After consumers *Alice a* and *Bob b* paying five dollars, then we use rules ($c4$) - ($c7$) to derive facts ($c8$) and ($c9$) that become *Alice's a* facts ($a4$) and ($a5$) and derive facts ($c10$) and ($c11$) that become *Bob's b* facts ($b2$) and ($b3$). Rules ($c4$) and ($c5$) are specialized cases for rule ($o1$), while rules ($c6$) and ($c7$) are specialized cases for rule ($o2$), shown in Section 5.2. The mutable facts ($c12$) - ($c14$) indicate a snapshot of current usage, display, and print counts collected from both *Alice a* and *Bob b*; they will be taken into summation by *Charlie c*.

– **SWRL (Ontologies + Rules)** denotation of license agreement:

- Content distributor *Charlie's c* site:

- Ontology:
  $hasDisplayRights \sqsubseteq hasUsageRights$
  $hasPrintRights \sqsubseteq hasUsageRights$
  $\leq (hasDisplayCount_{\{a,b\}}.eBook,\ hasUsageCount_c.eBook)$
  $\leq (hasPrintCount_{\{a,b\}}.eBook,\ hasUsageCount_c.eBook)$
  $\{Alice, Bob\} \overset{domain}{\longleftarrow} hasUsageRights \overset{range}{\longrightarrow} R_1,$
  where $R_1 = \leq_{10} hasUsageCount_c$
  $\wedge \geq_{2007/05/07/0900} hasDateTime_c.Time$
  $\wedge \leq_{2007/05/10/2400} hasDateTime_c.Time$
  $\exists =_\alpha \exists = sum(\exists \leq_5 hasDisplayCount_i.\{HarryPotter\}),\ i \in \{a, b\},$
  where α: $\exists hasDisplayCount_c.\{HarryPotter\} \longleftarrow (c1)$
  $\exists =_\beta \exists = sum(\exists \leq_1 hasPrintCount_i.\{HarryPotter\}),\ i \in \{a, b\},$
  where β: $\exists hasPrintCount_c.\{HarryPotter\} \longleftarrow (c2)$
  $\exists =_\delta sum(\alpha, \beta),$
  where $\delta : \exists hasUsageCount_c\{HarryPotter\} \longleftarrow (c3)$

- Rules:
  $hasDisplayRights(?x, ?r) \wedge hasSell_dRights(?x, ?r)$
  $\Longrightarrow hasDisplaySell_dRights(?x, ?r) \longleftarrow (c4)$

  $hasPrintRights(?x, ?r) \wedge hasSell_dRights(?x, ?r)$
  $\Longrightarrow hasPrintSell_dRights(?x, ?r) \longleftarrow (c5)$

  $hasDisplaySell_dRights(?x, ?r) \wedge delegate_g(?x, ?y)$
  $\wedge\ hasPrepaid(?y, ?a) \wedge \Longrightarrow hasDisplayRights(?y, ?r) \longleftarrow (c6)$

  $hasPrintSell_dRights(?x, ?r) \wedge delegate_g(?x, ?y)$
  $\wedge\ hasPrepaid(?y, ?a) \Longrightarrow hasPrintRights(?y, ?r) \longleftarrow (c7)$

- Facts:
  $eBook(HarryPotter)$
  $hasDisplayRights(Charlie, HarryPotter)$
  $hasPrintRights(Charlie, HarryPotter)$
  $hasSell_dRights(Charlie, HarryPotter)$
  $hasDisplaySell_dRights(Charlie, HarryPotter)$
  $hasPrintSell_dRights(Charlie, HarryPotter)$
  $\exists =_5 hasPrepaid(Alice)$
  $hasDisplayRights(Alice, HarryPotter) \longleftarrow (c8)$
  $hasPrintRights(Alice, HarryPotter) \longleftarrow (c9)$
  $\exists =_5 hasPrepaid(Bob)$
  $hasDisplayRights(Bob, HarryPotter) \longleftarrow (c10)$
  $hasPrintRights(Bob, HarryPotter) \longleftarrow (c11)$
  $delegate_g(Charlie, Alice)$
  $delegate_g(Charlie, Bob)$
  $\exists =_7 hasUsageCount_c(HarryPotter) \longleftarrow (c12)$
  $\exists =_6 hasDisplayCount_c(HarryPotter) \longleftarrow (c13)$

$\exists =_1 hasPrintCount_c(HarryPotter) \longleftarrow (c14)$

In the bootstrapping stage, all ontology statements, rules, and facts are described as license agreements and will be sent to $Alice\ a$ and $Bob\ b$ from $Charlie's\ c$. Facts $(a4)$ and $(a5)$ and facts $(b2)$ and $(b3)$ were previously inferenced on $Charlie\ c$ site via rule $(c6)$ and $(c7)$, where they were separately sent to $Alice\ a$ and $Bob\ b$. Each time $Alice\ a$ requests to display or print permission for $HarryPotter$, then associated rules $(a1)$ or $(a2)$ will be enforced to check whether conditions on the rule antecedents are all true. In fact, rules $(a1)$ and $(a2)$ are specialized cases of rule $(o3)$ in Section 5.2. For example, if $Alice\ a$ asks permission to print $HarryPotter$, her request will be granted because facts $(a5)$, $(a7)$, and $(a8)$ imply that all of the conditions on rule $(a2)'s$ antecedents are all true. Therefore, the conclusion $Permitted_a(Print, HarryPotter)$ is true. On the other hand, if $Bob\ b$ asks permission to print $HarryPotter$, it will not be granted because mutable fact $(b5)$ implies that $<_1 hasPrintCount_b(HarryPotter)$ is false, so the conclusion $Permitted_b(Print, HarryPotter)$ can not be derived. In our policy framework, we assume that what is not explicitly permitted is forbidden. Therefore, a permission request to print will be denied.

- Content consumer $Alice's\ a$ site:

  * Ontology:
    Similar to content distributor $Charlie's\ c$ site's ontology, except the usage rights constraints are local to $Alice\ a$

  * Rules:
    $hasDisplayRights(?x, ?r) \wedge <_{10} hasUsageCount_c(?r)$
    $\wedge <_5 hasDisplayCount_a(?r) \wedge \geq_{2007/05/07/09:00} hasDateTime(?t)$
    $\wedge \leq_{2007/05/10:24:00} hasDateTime(?t)$
    $\Longrightarrow Permitted_a(Display, ?r) \longleftarrow (a1)$

    $hasPrintRights(?x, ?r) \wedge <_{10} hasUsageCount_c(?r)$
    $\wedge <_1 hasPrintCount_a(?r)$
    $\wedge \geq_{2007/05/07/09:00} hasDateTime(?t)$
    $\wedge \leq_{2007/05/10:24:00} hasDateTime(?t)$
    $\Longrightarrow Permitted_a(Print, ?r) \longleftarrow (a2)$

  * Facts:
    $eBook(HarryPotter) \longleftarrow (a3)$
    $hasDisplayRights(Alice, HarryPotter) \longleftarrow (a4)$
    $hasPrintRights(Alice, HarryPotter) \longleftarrow (a5)$

$\exists =_1 \ hasDisplayCount_a(HarryPotter) \longleftarrow (a6)$
$\exists =_0 \ hasPrintCount_a(HarryPotter) \longleftarrow (a7)$
$\exists =_7 \ hasUsageCount_c(HarryPotter) \longleftarrow (a8)$
$hasDateTime_a(2007/05/09/09:00) \longleftarrow (a9)$

- Content consumer $Bob's\ b$ site:

  ∗ Ontology:
  Similar to content distributor $Charlie's\ c$ site's ontology, except the usage rights constraints are local to $Bob\ b$

  ∗ Rules:
  Similar to content consumer $Alice's\ a$ site's rules, except the condition's subscript is b in rules $(a1)$ and $(a2)$

  ∗ Facts:
  $eBook(HarryPotter) \longleftarrow (b1)$
  $hasDisplayRights(Bob, HarryPotter) \longleftarrow (b2)$
  $hasPrintRights(Bob, HarryPotter) \longleftarrow (b3)$
  $\exists =_5 \ hasDisplayCount_b(HarryPotter) \longleftarrow (b4)$
  $\exists =_1 \ hasPrintCount_b(HarryPotter) \longleftarrow (b5)$
  $\exists =_7 \ hasUsageCount_c(HarryPotter) \longleftarrow (b6)$

6 Discussion

In Fig 3, the XML-based rights expression languages (RELs), such as ODRL, XrML, and P3P, are convenient for automatic machine (or agent) processing but do not have formal semantics to represent and enforce access rights permission. Therefore, policies based on these RELs to describe a license agreement (or contract) are usually written in Natural Language to indicate their meaning for the verification of access rights permission. As a result, these natural language policies sometimes are open to interpretation, which result in ambiguity of policy semantics. In order to remove this problem, people use FOL to represent and reason access rights control policies (see Fig 3). As we know, FOL-based policies have a formal and clear syntax and semantics, even these FOL-based policies usually have to limit their expressive power in order to capture those license agreements that are originally written in English. Unfortunately, policies shown as FOL always require policy writers and readers to be logicians. Furthermore, policies indicated as a generic full FOL may feature compliance checking that may be undecidable for their computation time.

To resolve this dilemma, we are going to explore the expressive power of different FOL-based policies representations to decide which conditions allow us to have both decidable and enforceable semantics capability of rights delegation policies. In order to have a decidable and tractable fragment of FOL-based policies to enforce respective compliance checking, we usually restrict policies as datalog Horn rules, where they are negation-free, function-free, and with limited number of parameter parities. Description Logic (DL) is a decidable fragment of

FOL and Logic Program (LP) is closely related to the Horn fragment of FOL. In general, a full FOL is undecidable and intractable even under the datalog restriction. As shown in [5], Description Logic Programs (DLP) is an expressive fragment of FOL and it provides a significant degree of expressiveness and substantially greater power than the RDF-S fragment of DL. Based on DLP, the Semantic Web Rule Language (SWRL) is considerably more powerful than either the OWL DL ontology language or the datalog Horn style rule language alone because SWRL extends OWL with the basic kinds of datalog Horn rule, which states as predicates are limited to being OWL classes and properties with a maximum parity of two, etc [9].

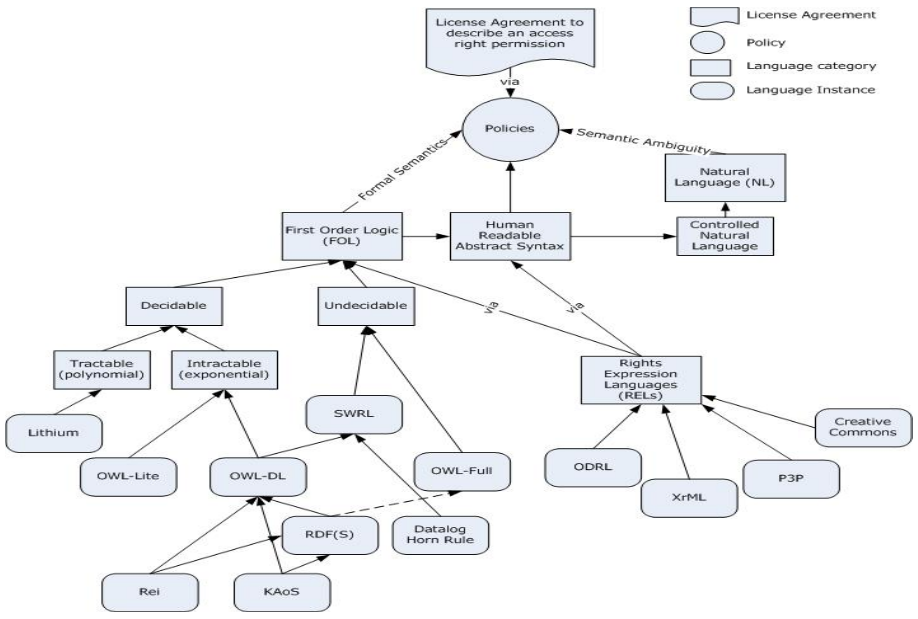

Fig. 3. A license agreement to denote access rights permission from a variety of policy language representations, such as Natural Language, Controlled Natural Language, First Order Logic (FOL), and Rights Expression Languages (RELs), etc

Policies in datalog Horn rules always assume that what is not explicitly permitted is forbidden. In that case, we can not distinguish forbidden access rights from unregulated access rights in a license agreement. Furthermore, we might need function capability in FOL-based policies to support translating English policies to FOL ones. Therefore, a tractable sublanguage, Lithium, with bipolars restriction, e.g., no bipolar literals in the FOL rules, was proposed in [8] to support its representation of denying policies and limited functions in their license agreement policies. Even though the Lithium policies are based on the relaxation of the datalog Horn rules, we still believe that this tractable policy language is only located somewhere in a small subset of FOL language. Therefore, it still lacks a large portion of OWL-DL and datalog Horn rule expressive power to serve both right delegation ontology and usage (or transfer) rights delegation rules, as shown in Section 4 and Section 5.2.

In [20], KAoS and Rei policy languages were shown as originally from DAML $\rightarrow$ OWL and RDF-S so it is quite trivial that these two policy languages are merely a subset of SWRL. Therefore, the expressive power of KAoS and Rei are less than SWRL because the rights delegation policies cannot be shown as a pure OWL-DL ontology language alone. In [12], Rei was extended to be a policy and delegation framework that includes inter-related resources, policies, policy languages, and meta-policies. However, authorization delegation policies were not explicitly seen in this study.

In this paper, we utilize the power of SWRL combined language to demonstrate the possibility of semantic-driven enforcement of rights delegation policies. A license agreement for a rights delegation policy is a policy set showing any number of prerequisites and relevant policies. A policy set is composed of facts, ontologies, and rules. These license agreements are distributed by distributor *Charlie* to consumers *Alice* and *Bob*. In this delegation scenario, the usage rights are applied to the entire *eBook* class instead of merely to the instance of *HarryPotter*'s *eBook*. In this policy-aware distributed DRM system, each trusted DRM node should faithfully enforce its rights delegation policies via its "compliance checking" inference engine.

There are several mutable facts in each node that express a prerequisite's dynamic status. These mutable facts will be updated and passed between distributor and consumers whenever a usage rights permission is granted and consumed. The mutable facts updating activity will be initiated as an Event-Condition-Action (ECA) reaction rule, where *event* might be triggered by a user's request or a message's arrival. The *condition* is specified in each relevant rights delegation rule and the *action* includes usage rights enforcement and mutable fact updating actions.

7 Conclusions

We have shown that the semantic formal model for an ODRL-based rights delegation policy can be enforced by expressing them as a combination of ontologies and rules. Based on ODRL's expressions and data dictionary, a rights delegation ontology is proposed in this study. Furthermore, we also express the rights delegation policy as a set of rules for usage and transfer (or duplicate) rights delegations. When verifying ODRL formal semantics, our SWRL approach is much more superior to the generic restricted FOL model because of the greater availability of a rights markup language and the higher expressive power of policy compliance checking from our SWRL language. A real usage rights delegation scenario is demonstrated in this paper to justify our formal semantic model.

References

1. Blaze, M., J. Feigenbaum, M. Strauss, Compliance Checking in the PolicyMaker Trust Management System, *Pro. of the Financial Cryptography 198.* LNCS, 1465, 1998, pp. 254-274.
2. Borgida, A., On the relative expressiveness of description logics and predicate logics, *Artificial Intelligence.* 82, 1996, pp. 353-367.
3. ContentGuard Inc. eXtensible rights Markup Language (XrML), Version 2.0.
4. Garcia, R., I. Gallego, and J. Delgado, Formalising ODRL Semantics using Web Ontologies, *2nd International ODRL Workshop.* Lisbon, Portugal, 2005.
5. Grosof, N. B., et al., Description Logic Programs: Combining Logic Programs with Description Logic, *WWW 2003.* Budapest, Hungary, 2003, pp. 48-65.
6. Guth, S., G. Neumann, and M. Strembeck, Experiences with the Enforcement of Access Rights Extracted from ODRL-based Digital Contracts, *DRM'03.* 2003.
7. Halpern, Y. J. and V. Weissman, A Formal Foundation for XrML, *Proc. of 17th IEEE Computer Security Foundations Workshop (CSFW'03).* 2003, pp. 251-263.
8. Halpern, Y. J. and V. Weissman, Using First-Order Logic to Reason about Policies, *Proc. of 17th IEEE Computer Security Foundations Workshop (CSFW'03).* 2003, pp. 187-201.
9. Horrocks, I., et al., OWL rules: A proposal and prototype implementation, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web 3.* 2005, pp. 23-40.
10. Iannella, R., Open Digital Rights Language (ODRL), W3C Note 19. September 2002, `http://www.w3.org/TR/odrl/`.
11. Kagal, L., T. Finin, and A. Joshi, A Policy Based Approach to Security for the Semantic Web, *ISWC 2003.* LNCS 2870, pp. 402-418, 2003.
12. Kagal, L., et al., Self-describing Delegation Networks for the Web, *IEEE Workshop on Policy for Distributed Systems and Networks, Policy 2006.* June 5-7, 2006.
13. LaMacchia, A. B., Key Challenges in DRM: An Industry Perspective, *DRM 2002.* LNCS 2696. 2003, pp. 51-60.
14. Li, N., B. N. Grosof, and J. Feigenbaum, Delegation Logic: A Logic-based Approach to Distributed Authorization, *ACM Trans. Information and System Security.* 6(1), pp. 128-171, 2003.
15. Lutz,C., Description Logics with Concrete Domains - A Survey, *Advances in Modal Logic.* Vol. 4, World Scientific Publishing Co., 2003, pp. 265-296.
16. Pan, J. and I. Horrocks, Web Ontology Reasoning with Datatype Groups, *ISWC 2003.* LNCS 2870, Springer, pp. 47-63.
17. Park, J. and R. T. Sandhu, The $UCON_{ABC}$ Usage Control Model, *ACM Transactions on Information and System Security.* 7(1), 2004, pp. 128-174.
18. Pucella, R. and V. Weissman, A Formal Foundation for ODRL, *Workshop on Issues in the Theory of Security (WITS).* 2004.
19. Stefik, M., Letting Loose the Light: Igniting Commerce in Electronic Publication, *Internet Dreams: Archetypes, Myths, and Metaphors.* MIT Press, 1996.
20. Tonti, G., et al., Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder, *ISWC 2003.* LNCS 2870, pp. 419-437, 2003.
21. Weitzner, D. J., et al., Creating a Policy-Aware Web: Discretionary, Rule-based Access for the World Wide Web, *Web and Information Security.* Idea Group Inc., 2006.