Persistent Access Control: A Formal Model for DRM*

Alapan Arnab & Andrew Hutchison
Data Networks Architectures Group
Department of Computer Science
University of Cape Town
Rondebosch, 7701
South Africa
{aarnab, hutch}@cs.uct.ac.za

ABSTRACT

Digital rights management (DRM) can be considered to be a mechanism to enforce access control over a resource without considering its location. There are currently no formal models for DRM, although there has been some work in analysing and formalising the interpretation of access control rules in DRM systems. A formal model for DRM is essential to provide specific access control semantics that are necessary for creating interoperable, unambiguous implementations. In this paper, we discuss how DRM differs as an access control model to the three well known traditional access control models – DAC, MAC and RBAC, and using these existing approaches motivate a set of requirements for a formal model for DRM. Thereafter, we present a formal description of LiREL, a rights expression language that is able to express access control policies and contractual agreement in a single use license. Our motivation with this approach is to identify the different components in a license contract and define how these components interact within themselves and with other components of the license. A formal notation allows for an uniform and unambiguous interpretation and implementation of the access control policies.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; D.4.6 [**Operating Systems**]: Security and Protection—*Access controls*

General Terms

Theory, Security

Keywords

Rights Expression Languages, REL, Access Control

1. INTRODUCTION

There are many definitions and interpretations attached to the words “Digital Rights Management”, or its better known acronym DRM. However, Rosenblatt et al.’s definition of DRM as technologies and techniques that provide “*persistent access control of digital data*” [34], is perhaps the most descriptive. This definition encompasses the requirements for both consumer DRM systems for protecting copyrighted works on the Internet and for enterprise DRM systems for protecting sensitive enterprise data.

For DRM to succeed, access control needs to be applied regardless of location of the data. However, traditional access control mechanisms have largely been restricted inside a defined boundary (such as an enterprise or a device) and this is the most significant difference between DRM and existing access control mechanisms. Furthermore, traditional access control approaches may not be entirely suitable for the purposes of DRM.

As discussed by Jajodia et al in [24], access control has two distinct parts, which are dependent on each other to function properly:

1. The means to represent the policies controlling access to a resource.

2. The means to implement the policies correctly and effectively.

Because the latter is almost a purely logical process, most access control models have been based on some sort of logical notation. In DRM systems, access control rules for an object are expressed in a *use license*. Use licenses are expressed in *rights expression languages (RELs)*, of which there are two major, general purpose RELs – MPEG REL based on XrML and ODRL [10].

In the XrML 2.0 specifications [1] the requirements for a REL are given as:

- *Comprehensive*: A language that shall be capable of expressing simple and complex rights in any stage in a workflow, lifecycle or business model.

- *Generic*: A language shall be capable of describing rights for any type of digital content or service (an ebook, a file system, a video or a piece of software)

- *Precise*: a language shall communicate precise meaning to all players in the system.

*This is the author’s version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published (will be published) in The Proceedings of the 7th ACM DRM Workshop, Co-Located with ACM-CCS 2007, Alexandria, Virginia, USA

DRM’07, October 29, 2007, Alexandria, Virginia, USA.
Copyright 2007 ACM 978-1-59593-884-8/07/0010 ...$5.00.

There are a number of criticisms of current REL implementations; particularly with regards to the expression of legal requirements when enforcing copyright [28, 12]. Mulligan et al. argues that RELs like XrML cannot be considered comprehensive until users are able to request additional rights [28]. They argue that this ability is crucial for the enabling of fair use. Felten on the other hand argues that DRM systems will never allow fair use since the languages cannot handle the expressions and the AI complexities in fair use [12]. In these respects, they argue that current RELs are not comprehensive.

Bechtold however argues that many of the XrML rules and definitions like rights transfers are not implemented in current DRM systems [7] and thus the failure of DRM systems to have fair use is not hampered by the language. Bechtold maintains that a suite of programs that can implement all the rules and definitions available in XrML will be able to achieve most of the requirements of DRM systems with less compromise from right holders [7]. This would require users to communicate with the right holder to request additional rights or changes in rights, as argued by Mulligan et al. In [3], Arnab and Hutchison detailed extensions to ODRL that would allow for bi-directional communications. The latest models for ODRL v2.0 incorporate some of the features discussed by Arnab and Hutchison [23].

While the above arguments have been in favour of extending the capabilities of RELs, Jamkhedkar et al. argued in [26], that there are a number of problems with current approaches to RELs. First, general purpose RELs have become too complicated, and by trying to address all the parts of DRM, they do not address any of the part completely. Although RELs are already modelled on access control models [28], Jamkhedkar et al. argued that there is no real definition of an access control model for DRM, and thus there is no mechanism to evaluate and inter-operate between different RELs. The authors promoted the need for a simpler model, encompassing a stateless, language-neutral, rights model; but did not present any rights model for DRM.

There have been no formal investigations into RELs from the vendors themselves. Halpern and Weissman detailed formalised semantics to XrML in [20], while Pucella and Weissman detailed a similar investigation into ODRL in [31]. Both papers investigated the current problems with interpreting use licenses, and the sources of ambiguities. Pucella and Weissmann also discussed a logic for reasoning about and interpretation of license agreements in [30]. In some respects, these investigations found that both XrML and ODRL were not necessarily precise in conveying their intended meaning.

While these investigations examined the interpretation of existing languages, they did not seek to put the languages on a completely formal base. In [19], Guth et al. did investigate the requirements for a contract language, and developed a contract schema (CoSa). In [18], Guth had a more comprehensive discussion on CoSa, the relationship between contracts and use licenses and the requirements for rights expression languages. While CoSa provided a formal structure, this approach does not extend to RELs and there is currently no formal description for any REL.

Thus, while both ODRL and XrML are generic in nature, and can cater for any type of digital resource [10], both can be considered to be non comprehensive and imprecise.

In this paper we present a formal model for DRM, by defining a persistent access control model in section 5, including a discussion on the requirements for an access control model for DRM in section 3, and refining the model through these requirements. Our motivation is to identify the various factors required to define the access control policies and how these factors interact between themselves (if there are multiple factors) and with the other factors in the policy. Our contribution differs from [20] and [31] in that we are creating an access control model from the ground up, while the previous contributions have examined existing languages and models before suggesting improvements.

In this paper, we also discuss how well existing and proposed RELs satisfy our model, but very briefly. Before we present our access control model for DRM, we briefly discuss existing access control models in section 2 and why they are not completely suitable for DRM. Before concluding, we discuss the interpretation and enforcement of our formal model in section 6.

2. EXISTING ACCESS CONTROL MECHANISMS

Currently, there are three widely accepted access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-Based Access Control (RBAC). In this section, we will briefly review these models, before we discuss how DRM differs from these models.

2.1 DAC, MAC and RBAC

In DAC based systems, access to data objects is restricted based on the identity of subjects and/or groups to which they belong [29]. Furthermore, in DAC based systems, a user with access to the protected data can delegate access to other users. DAC does have a number of specification levels, and criteria B3 to A1 requires implementors to allow creators to control the propagation of access delegation to other users (which would be necessary for DRM systems) [29], but as discussed by Reid and Caelli, modern operating systems implement a simpler version of DAC [32] which does not allow such controls. They argued that DAC based operating systems allow ordinary users of the system to define their own security. By granting ordinary users this ability, a user could reconfigure the security policy of the system to subvert the DRM protection. The authors also point out the inability of mainstream operating systems to support the principle of least privilege. Since system privileges are based on the users' identity, any program executing on behalf of a user is granted the same access control privileges as the user. There are no efficient mechanisms for restricting users' access control rights.

In MAC based systems, and in the associated Multi-Level security (MLS) systems proposed by Bell and LaPadula [9, 8], access control is assured through a central security administrator, and thus ordinary users of the system are prevented from reconfiguring the computer's security policy [33]. However, for the purposes of DRM, the rights holders (or the owners of the data) are not guaranteed any control over the consuming device.

In MAC based systems, access control is based on the user's credentials, with users classified under a hierarchical structure, discussed in the Bell-LaPadula model [8, 9]. The hierarchical structure allows greater rights for some users while allowing lesser rights for other users. Protected objects are classified under this structure, and the object does not determine the level of access for the user. This creates a problem for DRM systems, where it is sometimes necessary to determine access according to the nature of the object as opposed to the classification of the user. For example, the author of an article can be classified as a rights-holder and a reader. However, the classification of the author as a rights-holder means that he has access to other works that are not necessarily his own.

The third and newest, popular access control model is Role-Based Access Control (RBAC), first described by Ferraiolo and Kuhn in [15], and subsequently detailed further by Ferraiolo et al.

in [14], as well as Sandhu et al. in [35] and von Solms and van der Merwe in [39]. Ferraiolo and Kuhn argued that access to data should be determined by the function of the users in relation to the data, which are usually defined by roles users play in an organisation [15]. For this reason, a role-based access control model is more suitable than the DAC or MAC based approaches that were available at the time. von Solms and van der Merwe further argued that the role-based approach is a combination of the resource-based approach (as found in DAC) and the user-based approach (as found in MAC) [39].

A pure role-based approach is however not suitable for DRM, for two reasons. Firstly, a pure role-based approach may not be able to distinguish access depending on the function of the object (as opposed to the function of the user), which would create a problem similar to the author-user problem discussed earlier. Secondly, the definition of roles is determined by individual organisations, and these roles vary from organisation to organisation; and sometimes differ within organisations. This problem could be solved by defining access roles with respect to the organisation (or department); but this would severely restrict portability of sensitive data between organisations.

2.2 Differences between DRM and Existing Access Control Models

The main difference between DRM and traditional access control however, remains on the boundary of control. Traditional access control models operate on an object within a defined boundary: either a system or organisation. DRM however aims to operate on objects that do not have any defined boundaries, and thus across different systems and organisations.

The definition of the boundaries in existing access control models determine user management of these systems. Traditional access control models are strongly coupled with user management, and it is therefore possible to specify the complete range of resources accessible to a particular user or role.

Since DRM does not operate in a defined boundary, such a specification is not easily made. Instead, a DRM policy should aim to specify whether a particular user has access to a particular resource. It is possible for a producer to keep track of which users have access to a particular resource, but will never be able to track *all* the resources accessible by a particular user; unless the user management is completely bounded to the DRM system. However, it should be possible for DRM systems to have no direct relationship with user management; as long as the user management provider is trusted by the producer.

Role and group membership evaluation is also different in DRM systems. In RBAC and MAC, the enforcement mechanism has to decide whether a user is part of a specific group or fulfils a specific role. Alternatively, the enforcement mechanism can ask another system (usually the user management system) for help in making such a decision.

However in a DRM system, the consumer is not guaranteed to be online, nor can the consumer's device be trusted to make such a decision (since group and role membership is dynamic). Thus, the user management system has to provide proof of such membership. For this reason, hierarchies associated with RBAC and MAC user management are not directly relevant to DRM systems. The user management system associated with the DRM system has to cater for such functionality, but the DRM system itself does not care for the structure of such hierarchies. This is a significant difference in the functionality of DRM when compared to RBAC and MAC.

2.3 XACML and RELs

While RELs have had no formal foundations, this is not the case for other access control specification languages, where there are numerous contributions. In [27], Kudo and Hada described a formal model for a XML based access control language: XML access control language (XACL), which can be considered as a fore-runner to the OASIS standardised eXtensible Access Control Markup Language (XACML) [16].

XACML is a generic access control specification language, and should therefore be able to specify DRM use licenses. However, as discussed by Guth in [18], generic languages cannot guarantee reliability in covering all the requirements, although they do provide a high degree in flexibility.

Kudo and Hada identified the following elements in the primary policy definition of XACL:

- the *object*,
- the *subject* wishing to acccess the object,
- the *action* the subject wishes to perform and
- the *context* in which the subject wishes to perform the action.

Other schemes, such as the discussions by Jajodia et al. [24], Dai and Alves-Foss [11] and Ferraiolo et al. [13] use similar main components, with the addition of *roles* but without taking context into account. Arnab and Hutchison discussed in [4] that DRM use licenses are contractual agreements. Thus, use licenses need to be equivalent to contracts in form; and thus require additional information not required in existing access control specifications. We discuss these additional requirements next, in section 3.

3. ADDITIONAL REQUIREMENTS FOR RELS

DRM should not be seen as a mechanism to enforce copyright law but rather as a mechanism to enforce contracts on access and usage of digital data. In such a view, the primary role of a REL is not to express copyright but rather to express contractual agreements between the user and the rights holders regarding the terms and conditions of accessing the digital resource. This legal model applies to enterprise and consumer DRM systems. While contracts are generally formless, they do have certain features which need to be represented in a REL model; which is a formal representation of the syntax and semantics of the language.

1. **The Licensor:** Current access control policies only denote the subject of the policies. However, a contractual agreement requires the specification of the party that will provide the service or product, and the subject can in fact be anonymous. The licensor does not have to be the actual rights holder but rather a party that has been given the right to provide licenses to other parties.

2. **Agreement and Obligation:** A contract is an agreement between two or more parties, creating obligations for the parties to uphold [37]. Agreements give rise to obligations (and hence becomes a contract) when penalties are declared should a party not fulfill its part of the agreement; i.e. the licensee is penalised if they do not pay (the access to the work is removed for example) or the licensor is penalised if the quality of service promised is not delivered (the rights holder refunds the user for example). In [35], Sandhu et al. discussed

role-based access control models that specifically did not handle obligations, because of the complexities involved. The requirements specifications for ODRL v2 also has a discussion on the need to include obligations associated to contracting parties and individual permissions [22].

3. **Contract Constraints:** In [5] Arnab and Hutchison, detailed a few legal requirements with respects to contracts in DRM use licenses. Contract constraints are limitations applicable to the entire contract and not to a specific permission, like the number of devices that can be used by the user and the period of validity for a contract.

4. **Delegation of Rights:** A delegate can be defined as "*a person authorized to act as representative for another*" [2]. In rights delegation, the current licensor delegates the licensee the authority to act as a licensor to a set of other licensees. If possible, the licensor should be able to control every part of the use license for delegation. Delegation is important, as it allows the original rights holder to control delegation down the DRM value chain.

5. **Support for Third Parties:** Contracts can specify third parties, who can be appointed in various capacities such as mediators, monitors, escrow agencies etc. Note that third parties in the contract are not involved in the agreement itself.

As discussed earlier, in [26], Jamkhedkar et al. categorised different features available in current RELs, and proposed that most of these features be removed. We discuss these categories in detail below including their effect on the REL model.

1. **Authentication Protocol:** Authentication is a vital component of any access control model. While the protocol may not be necessary, an access control model will still require information relating to the identities for various users and resources. Furthermore, as DRM operates in a global space, these identities must also operate in a global namespace.

2. **Payment Mechanisms:** The terms of payment is an obligation, and thus must be expressible as such. Thus, there is no need to cater for payment as a separate component of RELs.

3. **Rights Enforcement:** This refers to the possible semantics implied by the REL (for example, if the right is to play a song once, when does the counter get incremented). Jamkhedkar et al. proposed a separation between the expression and interpretation of access control rights, and thus proposed that rights enforcement should not be coupled with rights expression.

4. **Content Tracking:** Quite a few technologies have been proposed to try content tracking as part of DRM. For this reason, RELs have developed mechanisms to express these requirements. Content tracking can be part of the description of the resource itself, or part of the terms and conditions. It does not need to be part of the access control model, but does need to be catered for.

5. **License Management:** License management refers to a number of issues, like the delegation of licenses and aggregation of licenses. As discussed earlier, delegation needs to be part of the access control model, but other functions such as aggregation could be removed.

6. **Negotiation Protocol:** The negotiation protocols for DRM discussed by Arnab and Hutchison in [5] are largely REL agnostic. However, some aspects of negotiation require support from the REL, but this is a matter of vocabulary and not the syntax and semantics of the REL. One exception is the attribute "*negotiable*", attached to every element of the offers, indicating whether the element is negotiable or not. We do not incorporate this attribute into the model directly, but needs to be incorporated in implementation. The default state of this attribute should be"*true*". Since a contract is also the end result of a negotiation process; and the REL must be seamlessly integrated with the negotiation protocols without imposing too much overhead.

While Jamkhedkar et al. have recently promoted a move to minimise the requirements for RELs, published requirements for ODRL are quite numerous [22], and most of these were published before their discussion. Some of these rights, such as delegation of rights have been discussed; but some REL specific requirements remain important. We have identified the following requirements, which have an impact on the REL model.

Req 1.2: ***Support the transfer of rights for content that is aggregated/dis-aggregated*** When there is delegation of rights, it may be necessary to delegate rights for only parts of a resource. The access control model needs to be powerful enough to express this.

Req 1.7: ***Provide a "NOT" expression*** In RELs, and in general, most access control models operate on granting access explicitly stated and denying any rights not stated. Jajodia et al. defined such policies as *open policies* [24]. In [22], the requirement for a "NOT" expression is motivated by the need to express agreements in the fashion of "allow rights for all actions except x". Such a requirement is analogous to *closed policies* defined by Jajodia et al. However, as we discuss in our model later, neither open nor closed policies make sense in the DRM space.

Req 1.8: ***Support rights and duties for all contract parties*** Obligations are not restricted to specific permissions, but could be tied to the entire agreement.

Some of the scenarios described as part of context by Kudo and Hadi [27] have been discussed above. The model for DRM we present here does not take any further contexts into account, and thus we do not include context as a separate entity in our model.

4. A LICENSING REL: LIREL

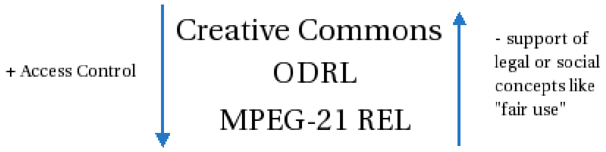

Figure 1: Comparing RELs functionalities [17]

In [17], Gonzalez commented that RELs can be seen in a spectrum, as shown in figure 1. At one end, there are RELs that try to represent copyright ideas and thus can express legal ideas such as fair use. On the other end of the spectrum are RELs that express access control rules. As we have previously discussed, DRM systems are not for the enforcement of copyright laws; but rather an enforcement of licensing terms and conditions, thus they are also a form of access control. For this reason, we think RELs should be able to express a legal contract, but at the same time, that contract should be able to define access control policies.

In the next section, we present a formal definition for a *Li*censing *R*ights *E*xpression *L*angauge or LiREL. This language is structured in the form of a licensing contract but is also able to express the access control policies required for DRM systems.

5. A FORMAL DESCRIPTION FOR LIREL

In this section we define a formal description for LiREL, using set notation. The notation is comparable to the notation used by Pucella and Weissmann in [30] and is also similar to the notation used by Ferraiolo et al. in [14] and in other access control specifications. A formal description provides specific semantics around access control, a necessary step for creating interoperable, unambiguous implementations. We also draw upon some of the requirements defined in the draft versions of ODRL v2 specified in [22, 23] to specify some of the requirements for individual elements.

As already discussed, a DRM use license is essentially a contract between two parties: the licensor and the licensee. A contract **C**, between two parties, *a* and *b*, any third parties involved (π) and the agreement α can expressed with the following tuple as:

$$\mathbf{C} = (a, b, \pi, \alpha)$$

For DRM use licenses, the contract needs to include details of the resources addressed in the license (*r*), the constraints for the contract (κ), and should be signed by representatives of the licensorsλ. Thus a DRM use license **L** can be expressed as:

$$\mathbf{L} = (C, r, \kappa) \| DSig_\lambda$$

This can be expanded, and (1) defines a DRM use license as:

$$\mathbf{L} = (a, b, \pi, r, \alpha, \kappa) \| DSig_\lambda \tag{1}$$

Note that there can be more than one signator for a license, and the licensor can also be a signator on a license. Also, it is not always necessary to have a signed contract, although it is considered to be good practice.

5.1 Language Semantics and Syntax vs Language Vocabulary

In this section we define the syntax and the semantics for a REL. In terms of natural languages, this is comparable to the definition of what is a noun, a verb or a pronoun etc., and how these can be combined to form a sentence. The vocabulary is the second component of the language, which allows for the expression of these terms. We do not focus on the vocabulary of LiREL; and while a standardised vocabulary is essential for interoperability, it will be difficult for us to define a comprehensive vocabulary for LiREL. For this reason, we focus solely on the syntax and semantics of LiREL and allow for the definition of the associated vocabulary separately. This approach is also used by ODRL and XrML although some comparisons of RELs tend to discuss the vocabulary of the languages [17, 36].

5.2 Obligations

Almost every term in a license can have obligations attached to it, and hence we begin our discussion with obligations, also referred to as duties by Guth [18]. As discussed earlier, some authors have felt that obligations add too much complexity to access control specifications [35], but obligations are a fundamental part of contracts, and thus necessary for DRM use licenses. Some obligations, like payment terms, can be enforced at machine level, but others are purely legal in nature; and disputes that arise will need to be resolved in arbitration or a court of law. Obligations can also have constraints attached to define limitations to obligations.

An obligation definition in a use license should contain the necessary details of the obligation, and should be able to specify whether the obligation is negotiable.

5.3 Constraints

Constraints define restrictions for specific terms in a license (such as the number of times an action is allowed), or apply to the license on the whole (such as the period of validity of the license). License constraints apply to every member of both *a* and *b*, but not necessarily to any delegated parties. Constraints can also be applied to obligations. As discussed in [23], constraints can be used to express mathematical terms, such as:

> number of pages is less than 100

where “number of pages” and “100” are operands and “less than” is the operator.

In addition to constraints, ODRL 1 also used a *condition* model [21]. Conditions allowed a license or permission to be invalidated once they became activated. Thus, it is possible to rewrite a condition as a constraint, and conditions were subsequently removed from the draft specifications of ODRL 2 [22].

All constraints applicable to the license, obligation or permission must be met before access can be granted. Constraints themselves can have further organisation to support different groupings. In such a case, the semantics of the constraint will depend on the definition of the constraint. Consider the following license constraints:

1. Valid Until: 2007-12-31
2. Device restriction:
   (a) device id: 8755GHGT876
   (b) device id: 867453HGT97

The licensee can only be granted access to the resource if the date of access is before 31 December, 2007 and the device that is used to access the resource has one of the listed device identities.

A constraint definition in a use license should be able to specify whether it is negotiable, and should be capable of expressing mathematical terms involving two operands and one operator.

5.4 The Licensors

a is the set of persons (natural or legal) who have been authorised by the rights holders of the resource, to license the access to resources *r* to prospective licensees. *a* does not need to be a comprehensive list, but must satisfy the following:

$$a = \{k_1 o_1, k_2 o_2 ... k_n o_n\}, \quad a \neq \emptyset, n > 0 \tag{2}$$

$$k \in a \Rightarrow \exists l \in r, k \in \text{ authorised licensors of } (l) \tag{3}$$

$$\forall l \in r, \exists k \in a, k \in \text{ authorised licensors of } (l) \tag{4}$$

We have provided a definition for *a*, in (2), every licensor (*k*) has an associated set of obligations *o*, with (3) providing a definition of a licensor while (4) also describes the relationship between the licensor and the resources. This definition of a licensor allows for a license to reference licensors who do not operate licenses on works referenced in the license, thus catering for licensing of compilations of works.

a should be interpreted as a list of licensors, and there is no relationship necessary between licensors. The rights holders should be referenced through a globally unique identity scheme. It should be noted that obligations for the rights holders could be purely legal in nature (for example, 24-hour telephonic support), and not enforceable on a computer system. However, this should be seen as a strength since it gives the use license a sounder legal grounding.

A licensor definition in a use license must specify the licensor's identifier.

5.5 The Licensees

b is the set of persons (natural or legal) or roles representing the consumers of the resources defined in *r*, defined in (5), while we define what is meant by a user in (6). Note, that unlike *a*, *b* can be defined as an empty set, and thus accommodate anonymous users. *b* should primarily be interpreted as a list of users who are given the permissions defined by α.

$$b = \{k_1 o_1, k_2 o_2 ... k_n o_n\} \quad n \geq 0 \tag{5}$$

$$k \in b \Rightarrow \forall l \in \ r, \text{k gets access to } l, \text{under conditions } \alpha \tag{6}$$

However unlike licensors, it should be possible to create relationships between the users (and roles). For example, it should be possible to define a user list as:

$$b = \{(Alice \wedge Journalist), (Bob), (Eve \wedge Teacher \wedge Mrs\ Smith)\}$$

Thus, unlike for *a*, the definition for *k* in *b* is no longer just a single identity, but rather a group of identities or roles as defined in (7). To meet (6), all the roles and identities comprising k_i must be satisfied for k_i to gain access to a resource. Individual identities/roles may also have obligations, as well as obligations that affect k_i as a whole.

$$k_i o_i = \{j_1 o_1 \wedge j_2 o_2 \wedge ... \wedge j_t o_t\} \cdot \ o_i \tag{7}$$

$$t > 0$$
$$0 \leq i \leq n$$

where *j* is a person, role or group associated with a identity scheme.

Like the licensors, licensees should be referenced through a globally unique identity scheme, preferably the same scheme used to reference licensors. Roles can be catered for by a credentials service, and although a global credentials service does not currently exist, setting up such a service should not be too difficult. Some identity systems, like Kerberos, already provide a credential service.

A licensee definition in a use license must specify the licensee's identifier and must be able to indicate whether the inclusion of the licensee in the use license is negotiable.

5.6 The Third Parties

π is the set of persons (natural or legal) who have been appointed as third parties by both *a* and *b* as part of the agreement α.

$$\pi = \{k_1 o_1, k_2 o_2 ... k_n o_n\} \quad n \geq 0 \tag{8}$$

Similar to licensees, π there could relationships between users and roles appointed as third parties. For example, third parties for an electronic contract could be:

$$\pi = \{(Verisign \wedge CertificateAuthority),$$
$$(Thawte \wedge CertificateAuthority), (John \wedge Judge)\}$$

Thus, as for *b*, the definition for *k* in π is no longer just a single identity, but rather a group of identities or roles as defined in (9). The entire third party set comprising of k_i must satisfy their obligations as part of the contract. Individual identities/roles for the third parties may also have obligations, as well as obligations that affect k_i in as a whole.

$$k_i o_i = \{j_1 o_1 \wedge j_2 o_2 \wedge ... \wedge j_t o_t\} \cdot \ o_i \tag{9}$$

$$t > 0$$
$$0 \leq i \leq n$$

where *j* is a person, role or group associated with a identity scheme.

Like the licensors and licensees, third parties should be referenced through a globally unique identity scheme, preferably the same scheme used to reference licensors and licensees.

A third party definition in a use license must specify the third party's identifier and must be able to indicate whether the inclusion of the third party in the use license is negotiable.

5.7 The Resources

r is the set of resources, which the licensees *b* are given access to under conditions α by the licensors *a*. Licensees are given access to all the resources identified in *r*. Like the licensors, licensees and third parties, resources need a globally unique identity scheme. Furthermore, the DRM system needs to be able to perform identity verification for resources; i.e. *establish the truth of a claimed identity* [38]. However, unlike identity systems for users, most identity systems for digital resources do not provide a verification service. Arnab and Hutchison discussed an identity system for digital resources that provide verification service in [6].

A resource definition in a use license must specify the resource's identifier and must be able to indicate whether the inclusion of the resource in the use license is negotiable.

5.8 The Agreement

The agreement α is the most important part of the use license, and defines the access control rules and policies. It is also the most complicated part of the model. α is composed of permissions (ρ), its constraints (κ), and obligations (o) associated with each of the permissions. Thus, the agreement can be defined as:

$$\alpha = \{\rho_1 \kappa_1 o_1, \rho_2 \kappa_2 o_2 ... \rho_n \kappa_n o_n\} \quad n > 0 \tag{10}$$

κ_i is a set of constraints applicable to the individual permission ρ_i. The interpretation and definition of κ_i will thus depend on ρ_i. Each permission ρ_i is part of a pre-defined permission set *PS*. The definition and interpretation of *PS* will differ according to the application of DRM, and is dependent ultimately on the implementation of the DRM controller (the system that interprets and implements the use license) and the rights holders, where the rights holders can choose a *PS* that is a subset of the entire set available from the DRM system. For example, in the traditional Unix file system:

$$PS = \{read,\ write,\ execute\}$$

In the Unix file system, a permission called "print" has no meaning, and thus cannot be enforced, even if it is expressed as part of the use license. The rights holders can choose to reduce *PS* to:

$$P\acute{S} = \{read, write\}$$

Thus, for a use license created with reference to $P\acute{S}$, *b* will always have the right to execute, even though the DRM controller can technically control that right. It can be argued that rights not defined in a PS should be unregulated instead of being allowed. The problem with this position however, is that DRM controllers can then choose to block rights that are unregulated (e.g. do not allow execute, even though execute $\notin \acute{PS}$). Removing blocks on unregulated permissions could become difficult for the licensee, and thus our position is to allow any rights not defined in a PS.

Using these examples, we can motivate a generalised set of conditions for enforcement of rights in (11). There is a difference in this model to the conventional view of DRM use license interpretation, where only permissions granted explicitly in the use license should be enforced (closed policy). In the third case of (11), the license should be considered invalid.

$$\begin{array}{lllr} \rho \in \alpha & \rho \in PS & \Rightarrow & \rho \text{ granted} \\ \rho \notin \alpha & \rho \in PS & \Rightarrow & \rho \text{ denied} \\ \rho \in \alpha & \rho \notin PS & \Rightarrow & \rho \text{ granted} \\ \rho \notin \alpha & \rho \notin PS & \Rightarrow & \rho \text{ granted} \end{array} \tag{11}$$

5.8.1 A More Complex Agreement

In the current definition of agreement (10), authorised users are given all the permissions present in α. However, there can be use cases where a more complex agreement is required. For example, it may be desirable to create an agreement for a PDF document as follows (ε represents the empty set of constraints/obligations):

$$\alpha = \{(view{\cdot}\varepsilon{\cdot}\varepsilon) \vee (view\ {\cdot}AdobePDFReader7{\cdot}\varepsilon \wedge print{\cdot}2{\cdot}\varepsilon)\}$$

This agreement can be interpreted as, the user has the right to view; or the user can view in the application AdobePDFReader7, and also get the right to print the document twice. There are many other use cases where a more complex definition for agreement may be necessary as detailed in (12), where $\odot$ represents the relationship between $\rho_i\kappa_i o_i$ and $\rho_{i+1}\kappa_{i+1}o_{i+1}$.

$$\alpha = \{\rho_1\kappa_1 o_1 \odot \rho_2\kappa_2 o_2 \odot \ldots \odot \rho_n\kappa_n o_n\} \tag{12}$$

However, using the distributive laws, it is possible to simplify (12) to

$$\alpha = \{\varrho_1 \vee \varrho_2 \vee \ldots \vee \varrho_m\} \tag{13}$$

where

$$\begin{aligned} \varrho_i = \{\rho_1\kappa_1 o_1 \wedge \rho_2\kappa_2 o_2 \wedge \ldots \wedge \rho_t\kappa_t o_t\} \\ \varrho \neq \emptyset \\ 1 \leq i \leq m \\ m > 0 \\ t > 0 \end{aligned}$$

Thus, an agreement can be defined as a set of non-empty permission groups, ϱ, with each permission group consisting of a non-empty set of permissions and their associated constraints and obligations.

5.8.2 The "NOT" Permission

The "not" permission expression was initially envisaged as a means to create easier agreements where the majority of the permissions (in a permission set) are allowed [22], and would thus allow open policies in DRM. This is however simply an easier expression mechanism at the tool level, and an unnecessary feature for the model itself. Furthermore, using the definition of the permission set and (11), the use of a "NOT" permission is almost meaningless. For this reason we do not have any specific support for a "not" function.

Thus, our model cannot be categorised in either of the traditional definitions of open and closed policies. Instead, through the use of permission sets, we implement closed policies on only a defined set of operations.

In DRM systems, the use of global closed policies can be disadvantageous to the licensee; as it leaves the possibility for the enforcement engines to enforce restrictions not specified as part of the rights package, or in an extreme case, allows the producers to update the rights enforcement engines to enforce restrictions not specified as part of the rights package at a future date. The use of a permission set protects the licensee from such abuse, in the present and the future; and the licensee is guaranteed that enforcement policies will not change without a change in the policy itself.

5.8.3 Delegation

Delegation is effectively a complicated version of ρ, but one which should probably be considered as a standard part of the LiREL model, instead of being part of a permission set. Delegation (δ) is really a modified version of **L**, and can be defined (where c is the delegated party) as:

$$\delta = (\acute{b}, c, \acute{\pi}, \acute{r}, \acute{\alpha}, \acute{\kappa})\ \ where\, \acute{r}\ \subseteq\ r,\ \ \acute{b} \subseteq\ b,\ \ c \nsubseteq\ b,\ \ c \nsubseteq\ a \tag{14}$$

Note that $\acute{\pi}$, $\acute{\alpha}$ and $\acute{\kappa}$ are not necessarily subsets of the current π, α and κ. This definition also means that it is possible to create infinitely long chain of delegation, as delegation can continue to be part of $\acute{\alpha}$.

5.8.4 Use License Specifications

A permission definition in a use license must specify the details of the permission and must be able to indicate whether the permission of the resource in the use license is negotiable.

5.9 Catering for Negotiations

As we discussed earlier, negotiations are the means to establishing a contract, but at the same time, as discussed by Jamkhedkar et al. there is a need to minimise the complexity of RELs [26]. Thus, it would be best, if it is possible to cater for negotiations without an increase in the terms of the LiREL model.

Illocution	Meaning
request(i,j,φ)	a request from *i* to *j* for a proposal based on φ
offer(i,j,φ)	a proposal of φ from *i* to *j*
accept(i,j,φ)	*i* accepts a proposal φ made by agent *j*
reject(i,j,φ)	*i* rejects a proposal φ made by agent *j*
withdraw(i,j)	i withdraws from negotiation with j

Table 1: Illocutions for a logic-based negotiation language as discussed by Wooldridge and Parsons

Wooldridge and Parsons discussed a logic-based language for bargaining-based negotiation in [40], which we have reproduced in table 1. In their specification, the final result of a successful negotiation is *accept(i,j,φ)*; which is very similar to our own specification of **L**.

Using the illocutions defined in table 1, and the definition of the final agreement in (1), we can define negotiations for DRM in table 2, which is equivalent to Wooldridge and Parson's definitions with φ = (π, r, α, κ).

From table 2, it is clear that, except for withdrawal, there is no real difference between each of the illocutions. In fact, each of these illocutions can be seen as a license state. Thus, we could cater for negotiations within LiREL without a substantial increase in the computational complexity of LiREL. For example, an offer could look like:

```
<license type="offer">
```

Illocution	Meaning
request(i,j,π, r, α, κ)	a request from *i* to *j* for a proposal based on α
offer(i,j,π,r, α, κ)	a proposal of α from *i* to *j*
accept(i,j,π,r, α, κ)	*i* accepts a proposal α made by *j*
reject(i,j,π,r, α, κ)	*i* rejects a proposal α made by *j*
withdraw(i,j)	i withdraws from negotiation with j
agreement(i,j,π,r, α, κ)	i concludes an agreement with j

Table 2: Illocutions for negotiation in DRM

Details of the offer

</license>

while an agreement would look like:

<license type="agreement">

Details of the offer

</license>

Other negotiation mechanisms such as auctions and bidding discussed in [5] can also be accommodated by increasing the vocabulary of the license states. Thus the complete set, as discussed in [5] would be:

- Agreement (the complete agreement, and the default state)
- Request
- Offer (can be used for both bargaining and bidding)
- Counter-Offer
- Accept
- Reject
- Tender

As discussed in the course of the individual license elements, each element other than the licensors can indicate whether they are negotiable or not. It does not make sense to allow licensees to negotiate the inclusion of licensors as part of the agreement.

5.10 Visual Model

From the definition of the model in this section, we can create a UML model of **L** as shown in figure 2. We have expressed *b* as a "licensee group" and π as a "third party group" to enable the more complex expressions which we described in section 5.5.

5.11 Comparison to Current RELs

There are a number of REL specifications available currently, but well known, standardised specifications of MPEG-REL (based on XrML) and OMA-REL (based on ODRL 1) are hardly used [26].

Even though RELs can be seen as an expression of access control [10], neither ODRL 1 nor XrML have formalised models for their specifications. Mulligan and Burstein did create a simple model for XrML in [28], as shown in figure 3, but it can not be considered a comprehensive model. However, it can be clearly seen from the model, that many of the features presented in our model, such as catering for duties and delegations, are missing from XrML.

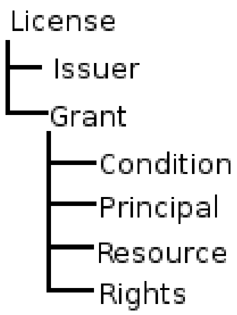

Figure 3: Simple XrML Model

ODRL 1 (base language for OMA-REL) does not have a formal model either, although Guth did map various components of ODRL 1 to the Contract Schema (CoSa) she developed in [18]. This was recognised, and the requirements specification for ODRL 2, does state the need for a formal specification for the language [22]. The latest draft of the ODRL 2 model specification does have a UML model, which is shown in figure 4. This model is very similar to our model shown in figure 2, although the ODRL model does provide a lot more detail.

There are a few differences between the ODRL model and the model we have presented in this paper:

1. **Prohibition and Permission:** ODRL has an explicit support for the "NOT" permission, although there could be confusion in the case where permission and prohibition are present in the license at the same time. As we discussed earlier, we do not believe this approach makes sense, and thus have not implemented such a mechanism in our model.

2. **The Legal Element:** The legal element provides some of the functions that we have generalised as "contract constraints"; but more specifically geared towards providing a firmer legal basis for DRM use licenses.

3. **Separation of Parties:** In our model, we have separated the parties into their respective functions with respect to concluding licenses, while the ODRL model makes no such distinctions. One advantage of our approach is to differentiate the interpretation of the relationship between the parties and the resources, which is not possible with the approach adopted for ODRL v2.

5.12 XML Schema

Due to space constraints, we cannot reproduce the XML Schema for our LiREL model. The reader is directed to the following website, http://pubs.cs.uct.ac.za/archive/00000411/, to download the XML schema, a sample vocabulary and examples. We have reproduced one of these examples in the appendix. We use the {read, write, execute} permission set, commonly used in file systems on Unix based operating systems.

6. ANALYSIS OF ACCESS CONTROL ENFORCEMENT

With a formal description for the use license, it is now possible to examine the second part of access control as defined by Jajodia et al. in [24]: the means to implement the policies correctly and effectively. In this section, we are going to use the formal model of LiREL to examine the issues related to the enforcement of the use license.

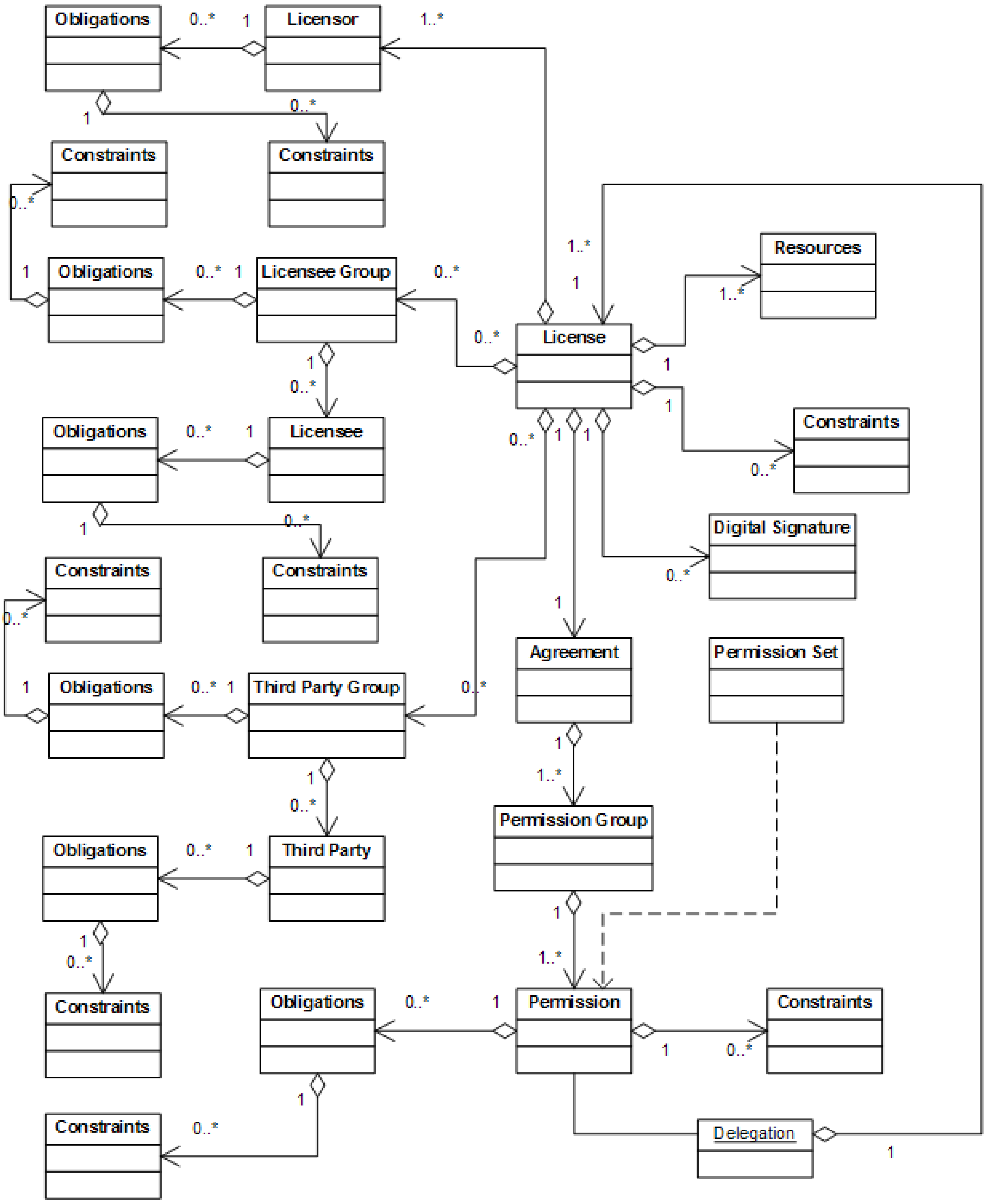

Figure 2: UML model of L

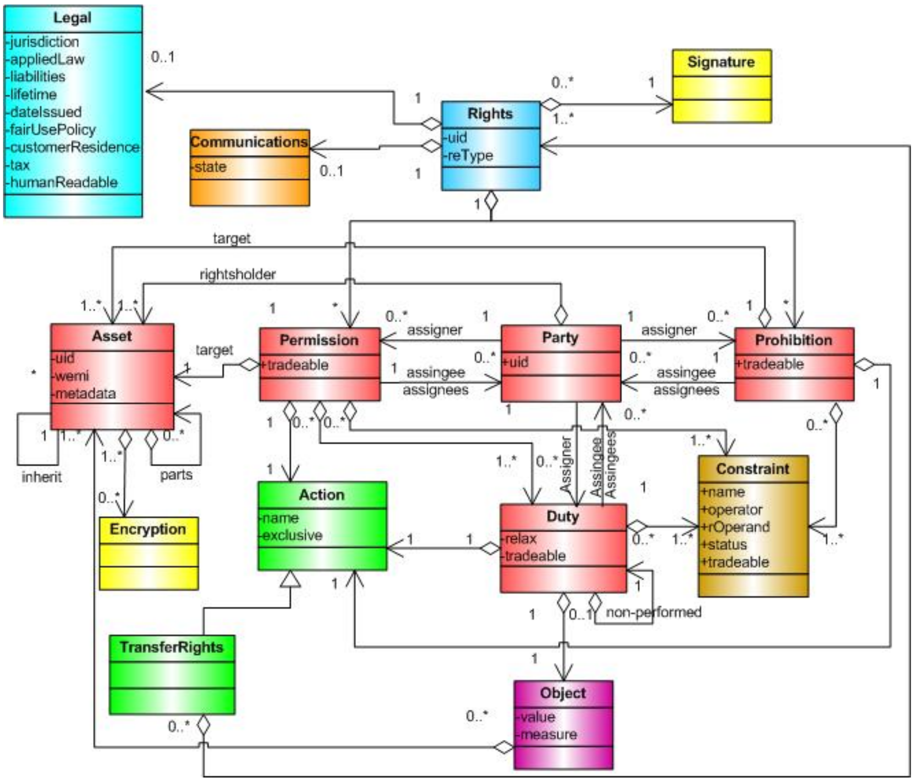

Figure 4: ODRL 2 Model (Draft) [23]

6.1 Validity of Use Licenses

Before a license can be enforced, the license needs to be valid. A license can be invalidated for three reasons:

1. It is malformed.
2. The constraints for the license cannot be met
3. The license is legally invalid.

6.1.1 Malformed License

A license is malformed if the license does not meet the specifications detailed in the model. For example, if the license does not state any licensors, or there are other syntax errors. A license can also be malformed if there are permissions granted by the agreement that are not present in the permission set associated with the agreement. This is the third case described in (11).

6.1.2 Constraints cannot be met

There are constraints that apply to the entire license, that can no longer be satisfied. For example, the time limit placed on the license may have expired, or the license may have stated that it could only be used a certain number of times. Once these constraints can no longer be satisfied, the license becomes invalid.

6.1.3 Legally Invalid

There are two main reasons why a use license can become invalid legally. First, the license could have been drafted by illegal means – for example, the licensor was not authorised to provide licenses. The second reason could be due to license revocation. Licenses can be revoked for a number of reasons, but there are two main reasons: the licensee can acquire a new license under different terms or one of the parties of the license could have broken the terms of the license.

6.2 Enforceability of Use Licenses

Even if a license is valid, it does not guarantee that the license is enforceable at the target device. A license is only enforceable, if the permission set associated with that license is enforceable by the device. The consumer cannot be granted access if a license is not enforceable.

It could be possible to allow selective enforcement of use licenses. In [25], the authors categorised rights enforcement in two levels: upper and lower level enforcement. In such a case, it would also make sense to separate enforceability into two parts. But such a use license should ideally make use of two distinct permission sets, one for each of the levels of enforcement.

6.3 Conflict Resolution

In [24], Jajodia et al. defined conflict resolution as the process undertaken when there are conflicting authorisations for the same subject. Jajodia et al. discussed three approaches to conflict resolution:

1. **No Conflict:** A conflict state indicates an error in the access control system.
2. **Denials take precedence:** A negative authorisation takes precedence over a positive authorisation.
3. **Permissions take precedence:** A positive authorisation takes precedence over a negative authorisation.

We propose the use of permission precedence for DRM systems. As long as the user can present a valid and enforceable use license for a particular action over a resource, the action should be allowed. Jajodia et al. also discussed the conflicts arising from different delegated authorisations. Permission precedence addresses this issue, and the other factors discussed by Jajodia et al. thus do not apply.

6.4 Deciding a Request

Decision on whether a user is granted an action on a resource is undertaken by the enforcement agent, or the DRM controller. As already discussed, given a request for permitting an action, the DRM controller will try to allow the action, from its known set of valid, enforceable licenses. More formally, the DRM controller will grant an action *req* to a consumer *k*, if

$$\exists \text{ a license l, such that } req \in \alpha,\ k\ \in\ b, \qquad (15)$$

and the following conditions are satisfied:

1. The obligations placed on the consumer are satisfied (assuming that the DRM controller can evaluate such obligations).

2. Any obligations attached to *req* are satisfied (assuming that the DRM controller can evaluate such obligations).

3. The constraints (if any) placed on *req* are met.

Note, from (11), *req* only needs to be evaluated if $req \in \acute{P}S$, where $\acute{P}S$ is the permission set enforceable by the DRM controller. If $req \in \acute{P}S$, then *req* needs to be evaluated against the permission set of the use license, as defined in (11).

6.5 Determining Cardinality

One of the features found in many access control models, especially RBAC models such as [13], is determining cardinality of the roles in a deployed access control system. There are possibly two summaries that rights holders would be interested in:

1. The number of consumers who have a certain permission on a certain resource.

2. The number of resources (and their associated permissions) attached to a particular consumer.

While both summaries are possible to calculate, they are processing expensive operations (taking account of revocations etc). Furthermore, if the licensors make use of external identity management services, the calculation becomes more difficult, as the cardinalities for roles may be much larger than than the number of licenses issued. While this may be inconvenient for rights holders and licensors, this is a privacy boost for licensees.

7. CONCLUSION

There is no formal description of DRM systems, including the specification and interpretation of access control policies. We believe DRM is another form of access control, and there are a number of differences between DRM and other well known access control models.

In this paper we presented a formal description of LiREL, a rights expression language that is able to express access control policies and contractual agreement in a single use license. Our formal description include:

1. The representation of the involved parties, individually or in groups. The three parties involved in a licensing agreement are the licensees, the licensors and third parties.

2. The representation of the resources covered by the license.

3. The details of the terms and conditions (the agreement) for access to the resources. Our model allows for the expression of multiple simultaneous conditions that need to be satisfied for access to be granted.

4. The representation of the constraints and obligations attached to individual parties and access terms. Constraints can also be attached to the entire license.

We also discussed the interpretation of LiREL, and the implications for the enforcement of DRM policies expressed in LiREL, including multiple conflicting licenses.

8. ACKNOWLEDGEMENTS

We would like to thank the members of the ODRL v2 working group, particularly Vicky Weissmann, for their comments and suggestions.

This work is partially supported through grants from the University of Cape Town (UCT) Council and the National Research Foundation (NRF) of South Africa. Any opinions, findings, and conclusions or recommendations expressed in this paper/report are those of the author(s) and do not necessarily reflect the views of UCT, the NRF or the trustees of the UCT Council.

9. REFERENCES

[1] *eXtensible rights Markup Language (XrML) 2.0 Specification*, 2001.

[2] AMERICAN HERITAGE DICTIONARIES, Ed. *The American Heritage Dictionary of the English Language*, fourth ed. Houghton Mifflin Company, 2000.

[3] ARNAB, A., AND HUTCHISON, A. Extending ODRL to Enable Bi-Directional Communication. In *Proceedings of the 2^{nd} International ODRL Workshop* (2005).

[4] ARNAB, A., AND HUTCHISON, A. Fairer usage contracts for DRM. In *Proceedings of the fifth ACM Workshop on Digital Rights Management, Co-Located with ACM CCS 2005* (2005), R. Safavi-Naini and M. Yung, Eds., ACM, pp. 1 – 7.

[5] ARNAB, A., AND HUTCHISON, A. DRM use license negotiations using ODRL v2.0, 2006. Submitted to the discussions in the 9th General Assembly of the Digital Media Project (DMP), Laussanne, Switzerland.

[6] ARNAB, A., AND HUTCHISON, A. Verifiable digital object identity system. In *Proceedings of the Sixth ACM Workshop on Digital Rights Management, Co-Located with ACM CCS 2006* (2006), K. Kurosawa, R. Safavi-Naini, and M. Yung, Eds., ACM.

[7] BECHTOLD, S. Digital Rights Management in the United States and Europe. IVir, Buma/Stemra - Copyright and the Music Industry: Digital Dilemmas.

[8] BELL, D. E., AND LAPADULA, L. J. Secure computer system: Unified exposition and multics interpretation. Mtr-2997 rev. 1, The MITRE Corporation. Online, last accessed: 2006-05-06.
URL: http://csrc.nist.gov/publications/history/bell76.pdf.

[9] BELL, D. E., AND LAPADULA, L. J. Secure computer systems: A mathematical model. *Journal of Computer Security 4*, 2/3 (1996), 229 – 263. Reprint of 1973 technical report M74 244, MITRE Corp.

[10] COYLE, K. Right Expression Languages, A report for the Library of Congress. Tech. rep., Library of Congress, USA, 2004.

[11] Dai, J., and Alves-Foss, J. Logic based authorization policy engineering. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics, and Informatics* (2002), pp. 230 – 238.

[12] Felten, E. Skeptical view of DRM and Fair Use. *Communications of the ACM 46*, 4 (2003), 57–59.

[13] Ferraiolo, D. F., Barkley, J. F., and Kuhn, D. R. A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security 2*, 1 (1999), 34 – 64.

[14] Ferraiolo, D. F., Cugini, J. A., and Kuhn, D. R. Role-based access control (RBAC): Features and motivations. In *Annual Computer Security Applications Conference* (1995), IEEE Computer Society Press. Available online: http://csrc.nist.gov/rbac/ferraiolo-cugini-kuhn-95.pdf.

[15] Ferraiolo, D. F., and Kuhn, D. R. Role-based access control. In *Proceedings of the 15th NIST-NSA National Computer Security Conference* (1992). Available online: http://csrc.nist.gov/rbac/ferraiolo-kuhn-92.pdf.

[16] Godik, S., and Moses, T., Eds. *eXtensible Access Control Markup Language*. OASIS, 2003. OASIS Standard; 18 February 2003.

[17] Gonzlez, R. G. *A Semantic Web Approach to Digital Rights Management*. PhD thesis, 2005. Online: http://rhizomik.net/%7Eroberto/thesis/Thesis.pdf.

[18] Guth, S. *Interoperability of DRM System*. Peter Lang, 2006.

[19] Guth, S., Neumann, G., and Strembeck, M. Experiences with the enforcement of access rights extracted from ODRL-based digital contracts. In *Proceedings of the 2003 ACM workshop on Digital Rights Management* (2003), ACM, pp. 90–102.

[20] Halpern, J. Y., and Weissman, V. A formal foundation for XrML. In *Proceedings of the Seventeenth IEEE Computer Security Foundations Workshop* (2004), pp. 251 – 263. URL: http://www.cs.cornell.edu/People/vickyw/papers-talks/XrML/CSFW04.pdf.

[21] Iannella, R., Ed. *Open Digital Rights Language (ODRL) 1.1*. IPR Systems Pty Ltd., 2002. URL: http://odrl.net/1.1/ODRL-11.pdf.

[22] Iannella, R., and Guth, S., Eds. *Open Digital Rights Language (ODRL) Version 2 Requirements*. 13 Feb 2005. URL: http://odrl.net/2.0/v2req.html.

[23] Iannella, R., and Guth, S., Eds. *ODRL V2.0 - Model Semantics*. 13 Jan 2007. URL: http://odrl.net/2.0/WD-ODRL-Model-20070113.html last accessed: 2007-08-23.

[24] Jajodia, S., Samarati, P., and Subrahmanian, V. A logical language for expressing authorizations. In *Proceedings of 1997 IEEE Symposium on Security and Privacy* (1997), pp. 31–42.

[25] Jamkhedkar, P. A., and Heileman, G. L. DRM as a Layered System. In *Proceedings of the Fourth ACM Workshop on Digital Rights Management* (2004), A. Kiayias and M. Yung, Eds., ACM, pp. 11 – 21.

[26] Jamkhedkar, P. A., Heileman, G. L., and Martinez-Ortiz, I. The problem with rights expression languages. In *DRM '06: Proceedings of the ACM workshop on Digital rights management* (New York, NY, USA, 2006), ACM Press, pp. 59–68.

[27] Kudo, M., and Hada, S. XML document security based on provisional authorization. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security* (New York, NY, USA, 2000), ACM Press, pp. 87–96.

[28] Mulligan, D., and Burstein, A. Implementing Copyright Limitations in Rights Expression Languages. In *Proceedings of the 2002 ACM workshop on Digital Rights Management* (2002), ACM.

[29] National Computer Security Center. A guide to understanding discretionary access control in trusted systems. NCSC-TG-003, September 1987.

[30] Pucella, R., and Weissman, V. A logic for reasoning about digital rights. *CoRR cs.CR/0405066* (2004).

[31] Pucella, R., and Weissman, V. A formal foundation for ODRL. *CoRR cs.LO/0601085* (2006).

[32] Reid, J. F., and Caelli, W. J. DRM, Trusted Computing and Operating System Architecture. In *Conferences in Research and Practice in Information Techology* (Newcastle, Australia, 2005), vol. 44, Australian Computer Society, Inc., pp. 127 – 136.

[33] Rhodes, T. Chapter 15 – Mandatory Access Control. FreeBSD Handbook, FreeBSD.org. Online, last accessed: 2006-05-06. URL: http://www.freebsd.org/doc/handbook/mac.html.

[34] Rosenblatt, B., and Dykstra, G. Integrating content management with digital rights management - imperatives and opportunities for digital content lifecycles. White paper, Giantsteps Media Technology Strategies, 2003. URL: http://www.giantstepsmts.com/drm-cm_white_paper.htm.

[35] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. Role-based access control models. *IEEE Computer 29*, 2 (1996), 38 – 47.

[36] Schmidt, A. U., Tafreschi, O., and Wolf, R. Interoperability challenges for DRM systems. In *IFIP/GI Workshop on Virtual Goods* (2004).

[37] Sharrock, R. *Business Transactions Law*, sixth ed. Juta & Co, LTD, 2002.

[38] Shirey, R. RFC 2828 – Internet security glossary, 2000. URL: http://www.faqs.org/rfcs/rfc2828.html.

[39] von Solms, S. H., and van der Merwe, I. The management of computer security profiles using a role-oriented approach. *Computers and Security 13*, 8 (1994), 673 – 680.

[40] Wooldridge, M., and Parsons, S. Languages for negotiation. In *Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI-2000)* (2000), W. Horn, Ed., John Wiley & Sons. http://citeseer.ist.psu.edu/wooldridge00languages.html.

APPENDIX

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
   The licensor creates the final agreement
-->
<lirel:license
   xmlns:lirel="http://www.cs.uct.ac.za/
       ˜aarnab/REL/lirel"
   xmlns:dd="http://www.cs.uct.ac.za/
```

```
      ˜aarnab/REL/lireldd1"
  xmlns:ds="http://www.w3.org/2000/09/
      xmldsig#"
  xmlns:enc="http://www.w3.org/2001/04/
      xmlenc#"
  xmlns:xsi="http://www.w3.org/2001/
      XMLSchema-instance"
  xsi:schemaLocation="http://www.cs.uct.ac.za/
      ˜aarnab/REL/lirel
  lirel.xsd http://www.cs.uct.ac.za/˜aarnab/
      REL/lireldd1
  lirel-lv1-dd.xsd" lirel:type="Agreement">

  <!-- Licensor -->
  <lirel:licensor>
    <lirel:identifier>
      jabber://licensor@exampleLiREL.net
    </lirel:identifier>
    <dd:qos>
      <lirel:detail>
        The license will provide access to a
        high resoultion (>300 DPI) version of
        the eBook
      </lirel:detail>
    </dd:qos>
  </lirel:licensor>

  <!-- Licensee -->
  <lirel:licenseeGroup>
    <lirel:party>
      <lirel:identifier>
        jabber://john@exampleLiREL.com
      </lirel:identifier>
      <dd:prePay lirel:negotiable="false">
        <lirel:detail>
           The licensee has to pay before an
           agreement is concluded
        </lirel:detail>
        <dd:money lirel:negotiable="false">
  <lirel:value>10</lirel:value>
  <dd:currency>EUR</dd:currency>
</dd:money>
      </dd:prePay>
    </lirel:party>

    <lirel:party lirel:negotiable="false">
      <lirel:identifier>
        credential://TrustedCredentials/teacher
      </lirel:identifier>
    </lirel:party>
  </lirel:licenseeGroup>

   <!-- Resources under discussion-->
   <lirel:resource>
     <lirel:identifier>
       vdoi://123.456/2/23/23.
     </lirel:identifier>
   </lirel:resource>

   <!-- Contract Terms-->
   <lirel:permissionGroup>
     <dd:read/>
   </lirel:permissionGroup>

   <dd:write>
     <dd:prePay>
       <lirel:detail>
          The licensee will pay an additional
          amount for the right to write.
       </lirel:detail>
       <dd:money>
         <lirel:value>10</lirel:value>
         <dd:currency>EUR</dd:currency>
       </dd:money>
     </dd:prePay>
   </dd:write>

   <!-- Contract Constraints -->
   <dd:jurisdiction lirel:negotiable="false">
     <lirel:value>
       Luxemburg
     </lirel:value>
   </dd:jurisdiction>

   <dd:validUntil lirel:negotiable="true">
     <lirel:value>
       2007-12-31
     </lirel:value>
   </dd:validUntil>

   <!-- Agreement Identifier -->
   <lirel:identifier>
      vdoi://123.456/1/45/23/Agreement
   </lirel:identifier>

   <lirel:dateOfIssue>
      2007-04-13
   </lirel:dateOfIssue>
</lirel:license>
```