

Digital Rights Management Implemented by RDF Graph Approach

Jin Tan Yang Huai-Chien Horng Mei Sheng Chao

National Kaohsiung Normal University

yangdav@nknuc.nknu.edu.tw

Abstract- This paper proposes a design framework for constructing Digital Rights Management (DRM) that enables learning objects in legal usage. The central theme of this framework is that any design of a DRM must have theories as foundations to make the maintenance, extension, or inter-operability easy. While a learning object consists of learning resource and its metadata, a DRM also needs metadata for describing itself as Rights Expression Language (REL). The proposed Resource Description Framework (RDF) graph design in this study is based on the Boolean operations of graph theory, where the RDF graph provides not only more coherent operations but also opportunities to maintenance and inter-operability at different platforms. Two algorithms for encoding and verifying rights in DRM are designed to deal with REL metadata in RDF format. This technological support also reduces the sophistication among role-assignments, learning objects, task ontology of DRM. The DRM module is embedded to SCORM-compliant Content Repository Management System (CRMS) for IPR (Intellectual Property Rights) protection. Finally, some implications of this study are also included.

Keywords: Digital Right Management, Role-Based Access Control, RBAC, Ontology, Resource Description Framework/RDF Schema (RDF/RDFS), Right Expression Language(REL)

1. Introduction

The digital revolution, powered by the engines of information and communication technologies (ICT), has fundamentally changed the way people think, behave, communicate, work and earn their living. It has restructured the means by which the world conducts economic and business activities and runs governments. Moreover, it has formed new ways to create knowledge, educate people and disseminate information.

Recently, the MIT Open CourseWare (OCW) has offered more than 700 free-charged courses, including lecture note, course material, examination, lecture video [18]. Thus, the OCW can be accessed by anyone eager to learn domain knowledge in the global village, free from any physical boundary. Although MIT has been a pioneering institution in learning technologies field, the real potential to keep

content providers who are willing to provide high quality of teaching materials is to protect their contents in legal usage. In this digital era, the Digital Rights Management (DRM) not only refreshes e-learning content, but also leads us towards a new model of education [20]. However, the enactment of DRM for content providers in education must rely on better algorithms to deal with Digital Rights metadata.

ICT has been adopted for access control in terms of protecting content providers rights. Many approaches have been proposed and applied in the real market such as Microsoft, Adobe, or IBM. Although those leading companies have digital rights to make their rewards, they only apply on specified customers, data formats, or delivering platforms. Thus, we must develop a new way for learning objects to be protected by access control. Generally speaking, most of the existing access control mechanisms implemented for web applications can be classified into two major categories—role-based access control (RBAC) models and the hypertext-based authorization models [17].

One of the core components of RBAC models is roles that represent different organizational responsibilities and functions. The use of the roles can simplify the task of authorization administration by organizing related access privileges to a role and assign users to the role [21]. Later, if a user is promoted to a new position in the organization, the user can simply be assigned to a new role and removed from the old one. In this study, we extend it to Sharable Content Object Reference Model (SCORM)-compliant learning objects or content packages [1]

Originally, a role in RBAC can access rights of the whole course package but not portions of the course package. This feature, however, seriously limits its applicability on the web. To overcome it, we adopt the Resource Description Framework (RDF) graph with Boolean operations to unite different learning objects as a unit. A RDF graph consists of nodes and arcs [3].

Content Repository Management System (CRMS), a collection of SCORM-compliant learning objects, has been developed [23]. It, however, did not consider DRM yet while it authorizes system administrator as gate keeper for assignment of digital

rights of learning resource. It violates content providers as primary authorization to confer rights for any potential users. In this study, we assume that only content providers can decide their contents to users with permissions.

Summarily, this study proposes a mechanism to enact the DRM for content providers within legal usage. To describe the authorization language, RDF graph is selected as Rights Expression Language (REL) language instead traditional XML-based language such as ODRL (Open Digital Rights Language) (ODRL, 2002), XrML (eXtensible rights Markup Language), or XACML (eXtensible Access Control Markup Language) [11]. The reason for the RDF graph being chosen is that RDF graph has superiority to deal with the complexity of rights assignment and Boolean operation for learning objects or content packages. Moreover, REL in RDF file formats can be used to reason by graph matching. In this study, content providers encode the digital rights in RDF file. Then, system will decode digital rights from RDF file while users access those learning objects or content packages in CRMS. With support of semantic RDF graph, this study proposes a simpler, machine processable, and extensible model for DRM.

In the following section, related literature on DRM will be reviewed and our solution will be proposed. Followed by literature review, the functionalities of RDF graph with Boolean operations will be investigated, and the implementations be presented. After the discussion and conclusion being presented, future studies will be recommended in the last section.

2. Literature Review

This paper proposes a design framework for constructing DRM that enables learning objects in legal usage. To reach the goal, some foundational theories such as DRM, ontology, and RDF graph with Boolean operations may offer the rationale.

2.1 Digital Right Management (DRM)

Although Digital Right Management (DRM) issue has been recognized as high ranking in cyber space where we are, it still has inclusive definitions so far. The Association of American Publishers points out that DRM consists of the technologies, tools and processes that protect Intellectual Property Rights (IPR) during digital content commerce, whereas Some [8][14][20] state DRM offers Intellectual Property (IP) Asset Creation/Capture, Management and Usage in terms of functionality. In other words, there are two purposes in DRM. One is that using ICT to protect IPR for rewarding the content providers. The other one is that both content providers and consumers can use legally by DRM system in which it reduces the conflict or chaos between content producers and content consumers.

Practically, DRM has three major components [6]:

(1) Expression – To describe the resource, ownership of the resource, and the terms and conditions of use such as Rights Expression Language (REL), closed security policy, agreements, or contracts.

(2) Authentication – To verify that the using the resource meets the rights associate to the resource such as Graph-based validation, Rule-based validation.

(3) Protection – To ensure only authorized users who are able to access by such mechanisms such as encryption. For example: Copy detection systems, digital signature, and information security Systems. This kind system often serves as infrastructure layer.

In this study, we focus on proposing a REL and a graph-based verification as rights enforcement of REL. Protection of information security technologies, however, is beyond the scope of this paper, it can be extended by REL via its extensibility.

2.2 The definition of Right Expression Language

Right Expression Language (REL) consists of three basic elements, which are rights, assets and parties [11]. First, rights are represented as expressions which grant certain usage or access permissions to digital goods or services. Comparing to rights, permissions can be more specified or restricted asset. Third, the party element might represent a legal role or physical person with unique ID.

To represent the real context in DRM, many REL, such as ODRL [19], XrML [22], ...etc, have been proposed and supported by different organizations. Traditionally, REL uses XML as representation language, such as ODRL, XrML, XACML and so forth. The XML data model is a text-markup oriented labeled tree. And as XML and XML Schema are designed primarily for fixed, tree-like documents, they are significantly inflexible for expressing meta-data of rights expression, which by its very nature is subjective, distributed and expressed in diverse forms. RDF, by contrast, has a very simple model consisting of labeled arcs that forms a graph-like data, and is also simpler, flexible to meet REL's requirements. Furthermore, any specific set of RDF statements forms a graph that can be serialized in XML and inherit its benefits.

In terms of rights verification, there is inclusive in both research and industry field. Traditionally, most systems use rule-based method and some others use closed encoded program to implement the inference of verification [12]. Those approaches make interoperability across applications and platforms difficult. To provide a more coherent and interoperability operations, RDF graph verification model adopted graph theory and its

operations to refine the task. This approach benefits mainly from shift partial inference logics to data-layer. Thus, this study adopts RDF graph as descriptor of REL to enhance its expression power by graph approach.

2.3 RDF graph/Ontology

An ontology is based on RDF/RDFS in terms of development stages [2]. Thus, we discuss RDF first. The foundation of RDF is based upon graph theory since it is a directed labeled graph in RDF/RFS representation [13]. To explain it, RDF/RDFS consists of three components such as subject, predicate, and object in sequential order. Second, an ontology is an explicit specification of a conceptualization [10]. In the context of AI, scholars describe the ontology as a program by defining a set of representational terms. Such definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Therefore, an ontology can be regarded as the statement of a logical theory that the first-order logic deals with propositions [9]. Subject and predicate in propositions are separately signified, reasoning whose validity depends on the level of articulation, and systems containing such propositions and reasoning.

To sum up, DRM must have theories as foundations to make the maintenance, extension, or inter-operability easy. RDF/ontology can be used to represent and infer among a set of vocabularies or propositions. Especially, the link between RDF graph and first-order logic makes inference feasible.

2.4 RDF and Graph Theory

In graph theory, a graph is composed by vertices and edges, denoted $G = (V, E)$ [5]. There are three basic Boolean set operations: "Union", "Intersection", and "Difference". While RDF graph, a derivation of graph theory, can also perform Boolean operations in terms of graph mapping. More precisely, RDF model can be used to construct different ontologies as graphs, and thus operations can be conducted as merging and disjointing among these ontologies [16].

3. Methodology

This paper proposes a design framework for constructing DRM that enables learning objects in legal usage. Two new approaches proposed in this study are: RDF graph REL representation and RDF graph verification. The former is given to provide more flexible and semantic ways to describe digital rights of learning resource. The latter offers a simplified verification mechanism.

3.1 Rights Expression Mechanism

RBAC model provides an easy way to clarify basic rights elements [17][21] by using roles, users, and permissions. Figure 3-1 shows three elements and their relationship. For example, Resources-Operation pairs form permission rights. On the one hand, the resource consists of objects such as Asset, SCO, or Content package. On the other hand, operations include view, download, or author, play.



Figure 3-1 : Mapping CRMS resources into RBAC model

In terms of task ontology [7][15], the role classification can be further divided as shown in Table 3-1 with instances of each classification. For example, user contribution has "Normal User", "creator", ... etc. Once content providers decide to release rights to everyone, they can set the role classification as others.

Role classification	Role instances
User Contribution	{[Normal User], [Creator]...etc}
User Identification	{ [Teacher], [Student]...etc}
Users' Discipline	{ [Chinese], [Math], [Science]...etc}
System Level	{ [Member], [Internal Staff],[Administrator]...etc}
Others	{ [Everyone]}

3.2 Data structure as RDFS graph

RDF Schema [3] is a mechanism that lets developers define a particular vocabulary for RDF data (such as 'hasRights') and specify the kinds of objects to which these attributes can be applied (such as 'Rights'). RDF Schema expressions are also valid RDF expressions – in fact, the only difference with 'normal' RDF expressions is that an agreement is made on the semantics of certain terms in RDF Schema and thus on the interpretation of certain statements. Objects can be declared to be instances of these classes using the type property. Constraints on the use of properties can be specified using domain and range constructs [4]. In this study, RDF Schema is used to encode REL ontology as machine-interpretable form for system operation.

The REL ontology shown in Figure 3-3 consists of five entities: Asset, Party, Permission, Constraint, and Requirement. The definitions of those entities are given as follows,

- (a). Asset: Attribute Sets: Asset_UID, and Asset_Type= {content package, Asset}. The Assets, a primitive learning object in SCORM definition, include any physical or digital content. Specifically, CRMS has two types of Assets such as SCORM-compliant content package and SCO. Those Assets must be uniquely identified. Any new type of Assets can be easily added/deleted through these two attributes.
- (b). Party: Attribute Sets= Party_UID, and Party_Type set = {Role, User}. Party might include individuals or rights holders with unique Party identification.
- (c). Permissions: Attribute Sets= {View, Download, Author...etc}. Permissions are the actual usages or activities of assets and content packages. In CRMS context, the Permissions include "View", "Download", "Author", and, "Play".
- (d). Constraint: Attribute Set= {usage count between 1 and n}. Constraints are limitations to access Permissions. For example, usage count constraint means the maximum number of times to access these assets.
- (e). Requirement: Requirements are the obligations needed to access the Permissions.

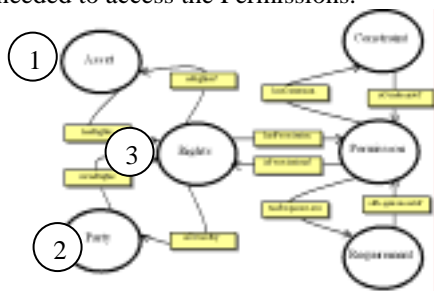


Figure 3-3: REL ontology graph based for CRMS

The REL ontology graph in Figure 3-3 can be encoded as RDF Schema in machine-readable format as shown in Table 3-2.

Table 3-2 : REL ontology described by RDFS (Partial)

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!ENTITY a 'http://protege.stanford.edu/system#'>
  <!ENTITY crms_drm
'http://protege.stanford.edu/crms_drm#'>
  <!ENTITY rdfs 'http://www.w3.org/TR/1999/PR-rdf-schema-19990303#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
  xmlns:a="&a;"
  xmlns:crms_drm="&crms_drm;"
  xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&crms_drm;Asset"
  rdfs:label="Asset">
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdf:Property rdf:about="&crms_drm;OwnRights"
  rdfs:label="OwnRights">
  <rdfs:domain rdf:resource="&crms_drm;Pa
```

```
<rdf:range rdf:resource="&crms_drm;Rights"/>
</rdf:Property>
<rdfs:Class rdf:about="&crms_drm;Party"
  rdfs:label="Party">
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
```

3.3 A scenario on encoding rights propositions in RDF graph

Based on the RDFS graph in the above section, each proposition can be represented as a RDF statement. Thus, it gives an intuitive semantics of precise thinking. For example, 'Teacher A' has Rights to 'View' the 'Learning Object' in condition 'c' and 'r', whereas 'Teacher B' has Rights to 'Download' on the 'Learning Object' in Figure 3-4. The propositions for Teacher A and Teacher B can be organized to represent rules and ontologies as shown in Figure 3-4. The proposition for Teacher A is "hasRights('Teacher A', Rights(isRightsof('LO'), hasPermission('View')))".



Figure 3-4: An example of encoding propositions in RDF graph

3.2 Rights Verification Mechanism

The verification engine is used as a reasoning engine and is consulted before every usage request or taking any action. Based on the answer of the verification engine, users might be allowed or rejected in terms of their requests. So verification engine plays an important role by ensuring rights execution in DRM. The main design concept of rights verification mechanism is that checking whether there are any inconsistencies between rights "Usage Request Graph" (URGr) and "Learning-Objects Rights Graph" (LRGr). By transforming usage request information into graph, it can be compared with graph of authorized rights information through Boolean operations of graphs. Access is allowed if two graphs are matched, and denied otherwise. The process of engine is shown in Table 3-5.

In Figure 3-5, RDF Instance Generator (RIG) gets information from users. The request from users can be represented as URGr. Thus, "Rights Validation Module (RVM)" can retrieve rights of learning object by Rights Instance Processor (RIP) and return a LRGr. In this section, Right Validator and RVM will be discussed in detail.

First, the algorithm for Right Validator is shown as follows,

Input : (1) "User request{User ID, Asset, Service}"
 (2) "Rights Instance Base"
Output : (1) "Access granted or denied"
 (2) "Insufficient rights information when denied."

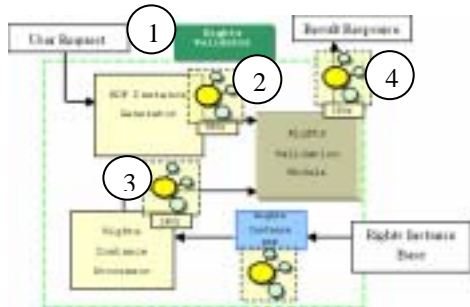


Figure 3-5: Framework of Rights Validator

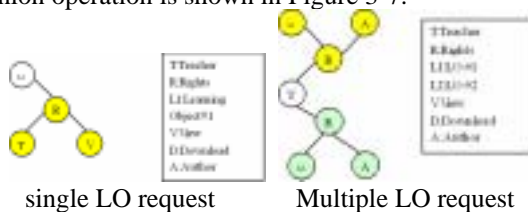
Second, RVM is a bridge between LRGr and RIP. Thus, URgr and LRGr must be designed before RVM mechanism.

URGr: An algorithm in Table 3-3 can be explained in how URGr to be created by RIG in Figure 3-5. Single LO and multiple LOs requests are explained in Figure 3-6 respectively. RIG generates the two URGr in both sides of Figure 3-6. The former is that a teacher calls for viewing learning object#1, while the latter calls for two learning objects simultaneously.

The algorithm of URGr is shown as follows,

Table 3-3: Algorithm : Generation of Usage Rights Graph
Input : "User Request{User ID, Asset, Service}"
Output : "Usage Rights Graph (URGr)"
Declare URGr as RDF_Graph FOR each requested(LO) URG.CreateStatement(Asset,hasname,requested(LO)) END FOR URG.CreateStatement(Right,haspermission,requeste d(permission)) URG.Createstatement(Party,role, requested(user_name)) RETURN URGr

LRGr: While most users will take many learning objects at one time, to verify the permission, multiple learning objects from RDF file should be united before sending it to RVM. The algorithm of "Union operation" is shown in Table 3-4. In other words, LRGr' = LRGr1 \cup LRGr2. An example for union operation is shown in Figure 3-7.



single LO request Multiple LO request

Figure 3-6:RIG generates URGr

Table 3-4: An algorithm: RIP generates LRGr by Union operation

Input : "Rights Instance Base"
Output : "Learning-object Right Graph (LRGr)"
Declare LRGr, LRGr' as RDF_Graph(s) READ all Rights_Instance_RDF from LO_base as LRGr' IF requested(LO).quantity > 0 FOR each LRGr LRGr = Union(LRGr, LRGr') END FOR ELSE LRGr = LRGr' END IF RETURN LRGr

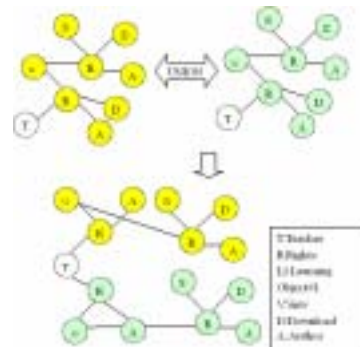


Figure 3-7 : An example of generating LRGr by RIP

Finally, the main design concept of rights verification mechanism is that checking whether there are any inconsistencies between rights ontology graphs. Thus, RVM is in charge of matching between LRGr and URGr by RDF graph. To get the difference, the "Difference" is applied. The algorithm is represented in Table 3-5.

Assume IRGr = URGr - LRGr.
 If IRGr= then "Access Granted"
 Else "Access Denied"

Table 3-5: Algorithm : Difference of URGr to LRGr
Input : "URGr", "LRGr"
Output : "Result of Validation", "Insufficient Rights Graph (IRGr)"
Declare IRGr as RDF_Graph REQUEST URGr, LRGr IRG = Difference(URGr, LRGr) CASE IRGr is : RETURN messege("Access Granted") CASE IRGr is not : RETURN messege("Access Denied") & IRGr ENDCASE

For example, the teacher A owns view right on learning object #1 in ① of Figure 3-8. Also, the LRGr of learning object #1 is constructed in ② of Figure 3-8. The result of matching ① and ② is ③ of Figure 3-8. In this case, the access right is granted while IRGr is empty.

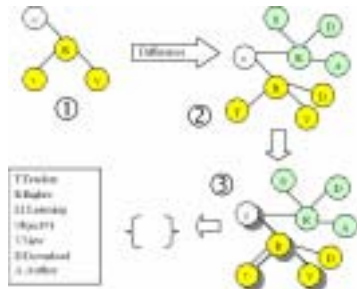


Figure 3-8: Access right is granted: Difference between URGr and LRGr is empty

In the same vein, if difference between URGr and LRGr is not empty, then the access right is rejected as shown in Figure 3-9.



Figure 3-9: Access right is denied: Difference between URGr and LRGr is not empty

3.5 Software Tools: Jena API

Jena is a Java API which can be used to create and manipulate RDF graphs. It is also originally developed at HP Labs Semantic Web Program. In this study, the Boolean operations of RDF graphs are implemented by calling Jena API methods. Thus, this study can deal with only system level, not finer level such as coding in RDF implementation.

4. Implementations

This section will demonstrate functionality by scenarios. The scenario presented here are encoding of learning objects in RDF format on CRMS and a verification process while users try to get any learning objects.

4.1 Encoding of learning objects in RDF format

CRMS has portal for users to login as shown in Figure 4-1. Assume a content provider, called “jimhorng”, authorizes a user, named “Koach”, a learning object with rights: “view” and “download”. After he follows the procedures to upload his SCORM™ learning object, the user graphic interface turns to Figure 4-2. jimhorng is asked to fill the five slots in Figure 4-2. Most of slots are menu-driven for him to choose instead of inputting.



Figure 4-1: CRMS portal

For example, he assigns an asset, called “jimhorng_15_3” to Kaoch by rights constraint consisting of ‘view’ and ‘download’. The system will automatically generate a unique ID, called “jimhorng_0_50_17” for the content. The RDF file is encoded in Figure 4-3. Once Koach wants to retrieve “jimhorng_15_3”, the RDF file will be parsed and generate as LRGr as shown in the above section. If URGr from Koach is matched to LRGr, then Koach can ‘view’ or ‘download’ the learning object. The GUI of Figure 4-2 is also allowed jimhorng to change his authorization to individuals or group.

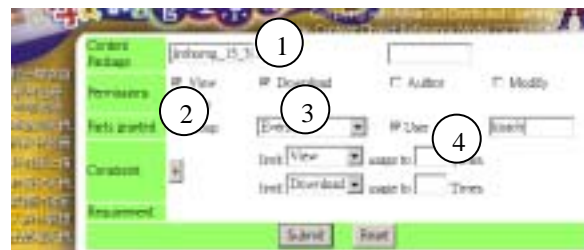


Figure 4-2: The GUI of authorized encoding on CRMS

```

<rdf:Description
  rdf:about="http://crms.edu.tw/#Permissions" 2
>
  <j.0:Permission_type>View</j.0:Permission_type>
  <j.0:Permission_type>Download</j.0:Permission_type> 3
  <vcard:FN
    rdf:resource="http://crms.edu.tw/#Constraint" />
</rdf:Description>
<rdf:Description
  rdf:about="http://crms.edu.tw/#Asset">
  <j.0:hasRights
    rdf:resource="http://crms.edu.tw/#Rights" 1
  />
  <j.0:Asset_name>jimhorng_0_50_17</j.0:Asset_name>
  </rdf:Description>
  <rdf:Description
    rdf:about="http://crms.edu.tw/#Party">
    <j.0:Role>koach</j.0:Role> 4
  </rdf:Description>
</rdf:RDF>
    
```

Figure 4-3: Rights assignment of Figure 4-2 encoded by Jena API in RDF format (Partial)

4.2 Verification mechanism

CRMS offers a function to allow users to search for learning object or content package by menu-driven keywords or intelligent agents. In Figure 4-4, Koach has found 10 learning object for assembling his content package. While he chooses learning object #10, three functionalities such as ‘author’, ‘view’, and ‘download’ are given to Koach. If he clicks “author”, then he will be rejected to do

so. Conversely, he can do the others because LRGr and URGr in RDF graph are matched.

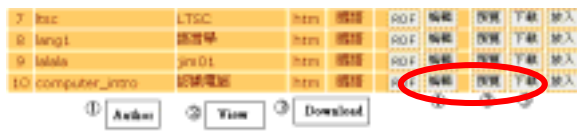


Figure 4-4: Three functionalities : author, view, and download on CRMS



Figure 4-5: Access right is granted

While Koach tries to view the learning object, he has been noticed that the system is validating his request. Later on, Figure 4-5 show that Koach has been granted to view the learning object.

Conversely, if Koach tries to author the learning object in Figure 4-6 and he is rejected while content provider did not authorize him to author on line in Figure 4-7.



Figure 4-7: Access right is denied

4.3 Comparisons between RDF graph and rule-based approach for DRM

In traditional artificial intelligence, most scholars choose XML as REL and adopt rule-based approach to verify mechanism in DRM. This study proposes a new approach by RDF as REL and RDF graph approach to verify DRM mechanisms of SCORM-compliant learning objects or content package. The RDF approach is superior to rule-based approach with at least three advantages, which are summed up as follows:

- (a). Simplification complexity of DRM: RDF as REL to represent DRM is based on graph theory and first-order logic. Thus, a programmer can encode and verify in RDF approach simpler than of rule-based approach. Actually, a RDF graph contains a set of propositions as ontology consists of a set of rules for verification. Through Boolean operation, both encoding and verifying are simplified. Moreover, the separation between encoding and decoding processes in this study is to reduce the complexity of DRM tasks.
- (b). Machine inter-operability in RDF graph: RDF graph consists of a set of rules. It means that those rule logics are also put in data layer. Thus,

a RDF API program can parse RDF files to construct RDF graph. In other words, computer program just needs to do Boolean operations of RDF graph instead to check all rules encoded at rule-based programs. Moreover, RDF file can be shifted to different platforms or programming languages while RDF parser has standard API. Conversely, the rule-based approach is tightly coupled with specified platform or programming language.

(c). Adaptation to different contexts: While new rules are added to RDF file or graph, there is no any change in inference engine. It reduces the burden of maintenance of computer system.

5. Discussion

Learning objects or content packages access control decisions are still always identified by RBAC. Such access rights to learning objects is granted or rejected according to user's personal information, such as account name and password. Once they enter to Learning Content Management System (LCMS) or LMS (Learning Management System), they can navigate, search, view, or download any learning object they want. It keeps content providers from supplying their learning objects to content repository continuously if they feel out of control on their content packages or SCOs. Therefore, digital right management should to assure in finer levels such as who owns authorization to access a learning object. Indeed, each content provider should own the whole rights to assign their contents to be shared with specified group or individuals.

Authorization schemes should be given by content providers while learning objects or content packages are uploaded. Verification schemes of accessing a content package should be also examined. The RDF as REL has been adopted as task ontology for encoding and verifying users' rights. If a user requests a learning object, the digital rights of learning object by content providers will be generated; the comparison of two digital rights between content providers and users' access right is compared with Boolean operations in terms of RDF graph. A user to be granted or rejected to access a learning object or content package is based on whether RDF graph given by content provider and users' request is matched.

6. Conclusions

This paper so far has focused on the design of DRM at CRMS. Certainly, a sound algorithm design by graph theories is critical for maintenance, interoperability in a LCMS. However, the ultimate goal of this study is to provide meaningful authorization for content providers and to verify the users' requests for specified learning objects or content packages. Therefore, the evaluation of DRM at CRMS is essential for future research. In the near

future, we will design a field assessment for performance and user satisfactions of RDF graph approach for DRM.

Finally, the success of technology-based DRM does not solely rely on technology. Considerations of human cognition, and the social context of that thinking, take precedence over technology. To this aim, people must have consensus on digital rights in the real social context.

7. Reference

1. ADL (2004). Sharable Content Object Reference Model (SCORM) 2004 2nd Edition. Retrieved August 20, 2004, from <http://www.adlnet.org/index.cfm?fuseaction=SCORMDown>.
2. Berners-Lee T., J. Hendler, & O. Lassila. (2001). The Semantic Web, *Scientific American*, 284 (5), p. 34-43, May 2001
3. Brickley, D. and Guha, R. (2000). Resource Description Framework (RDF) Schema Specification 1.0. Recommendation, *World Wide Web Consortium*. Retrieved August 20, 2004, from <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>.
4. Broekstra, J, Kampman, A. & van Harmelen, F. (2003). Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. In *D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, Semantics for the WWW*.
5. Diestel, R. (2000). *Graph Theory*. (2nd ed), Springer: Germany.
6. Downes S., Babin G., Belliveau L., Blanchard R., Levy G., Bernard P., Paquette G., Plourde S.(2003). Distributed Digital Rights Management: The EduSource Approach to DRM. *Proceedings of the First International ODRL Workshop* (Eds. Renato Iannella & Susanne Guth) 22–23 April, 2004, Vienna, Austria
7. Fensel, D., Motta, E., Decker, S. & Zdrahal, Z. (1997). Using Ontologies For Defining Tasks, Problem-Solving Methods and Their Mappings. In *E. Plaza et al. (eds.), Knowledge Acquisition, Modeling and Management, Lecture Notes in Artificial Intelligence (LNAI) 1319*, Springer-Verlag, 1997.
8. Fetscherin, M. & Schmid, M.(2003). Comparing the Usage of Digital Rights Management Systems in the Music, Film, and Print Industry. *ICEC 2003*, supported by ACM. Pittsburgh, PA.
9. Grau, B.C.(2004). *A Possible Simplification of the Semantic Web Architecture*. WWW2004, New York, USA.
10. Gruber, T.R.(1993). A Translation Approach to Portable Ontologies, in *Knowledge Acquisition*. 5(2), 1993, 199-220.
11. Guth, S.(2003). Rights Expression Languages. In Eberhard Becker, Willms Buhse, Dirk Günnewig, Niels Rump (Eds.), *Digital Rights Management: Technological, Economic, Legal and Political Aspects* (pp. 101 - 112). Heidelberg, Germany: Springer.
12. Guth, S., Neumann, G., Strembeck, M.(2003). Experiences with the Enforcement of Access Rights Extracted from ODRL-based Digital Contracts. *Workshop-Proceedings of the ACM Workshop on Digital Rights Management*, Washington D.C.
13. Hayes, P. (2003). RDF Semantics. *World Wide Web Consortium*, Retrieved August 20, 2004, from <http://www.w3.org/TR/2003/PR-rdf-mt-20031215/>.
14. Iannella, R. (2001). Digital Rights Management (DRM) Architectures. *D-Lib Magazine* Vol. 7(6).
15. Ikeda, M., Seta, K., Kakusho, O. & Mizoguchi, R. (1998). Task ontology: Ontology for building conceptual problem solving models. *13th European Conference on Artificial Intelligence ECAI'98*
16. Jannink, J.(1999). Ontology Maintenance with an Algebraic Methodology: a Case Study. In *Proceedings of 1999 AAAI workshop on Ontology Management*, Orlando FL, July 1999.
17. Lu, J. L. & Chen R. F.(2003). Design and implementation of a fine-grained menu control processor for web-based information systems. *Future Generation Computer Systems*, 1105–1119, MIT Press.
18. OCW (2004). The MIT OpenCourseWare Update: July 2004. Retrieved July 7, 2004, from <http://ocw.mit.edu/OcwWeb/Global/AboutOCW/newsletter.htm>
19. ODRL.(2003). *Open Digital Rights Language*. Retrieved July 7, 2004, from WWW: <http://odrl.net/>
20. Rosenblatt, B., Trippe, B. & Mooney, S. (2001). *Digital Rights Management: Business and Technology*. (1st ed.) New York: John Wiley & Sons.
21. Sandhu, R. S., Coynek, E. J., Feinstein, H. L. & Youmank, C. E. (1996). Role-Based Access Control Model. *IEEE Computer*, 29(2), p. 38-47.
22. XrML (2002). *Extensible Rights Markup Language Specification*. Retrieved July 7, 2004, from WWW: <http://www.xrml.org/>.
23. Yang, J. T. & Tsai, C. Y. (2003). An Implementation of SCORM-compliant Learning Content Management System-Content Repository Management System, ICALT2003. Athens: Greece, 2003.