

Extending ODRL to Support Dynamic Digital Rights Management

Dongyang Xu^{1,2}, Zhi Tang¹, Yinyan Yu¹ and Cheng Qu¹

¹ Institute of Computer Science and Technology, Peking University,
Beijing 100871, China
{xudongyang, tangzhi, yuyinyan, qucheng}@pku.edu.cn

² China Electronics Standardization Institute,
Beijing 100007, China

Abstract: The use license often plays the role of an agreement between the user and the copyright owner. Rights expression language should provide a description of interactive mechanism and license negotiation to avoid the ambiguous transformation between REL and authorization protocols. By extending the ODRL to establish a digital rights expression model of appending, updating and aggregating licenses, the interactive license management mechanism can be described. Some detailed examples of the extended model are also listed in this paper. The proposed scheme can support fine-grained authorization, progressive or updating authorization and combination authorization to achieve dynamic rights management of digital content and provide authorization on demand services.

Keywords: digital rights management, ODRL, license, authorization management.

I. Introduction

The use license is a key component of digital rights management [1-3] technologies. It is usually described by Rights Expression Language to define the rights of resources owned by the parties. The resources can be e-books, video, audio, images, software or other digital content. Generally, there are two categories of rights expression languages: one is based on logic language, such as LicenseScript [4]; the other is based on XML, such as XrML and ODRL [5-6], which are the widely used languages.

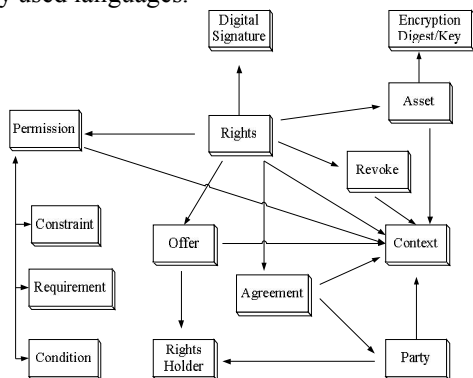


Figure 1. ODRL core model

ODRL Initiative is responsible for the development and management of ODRL, with the latest release version 1.1. Through the establishment of a descriptive model and adopting the syntax of XML, ODRL has defined some core entities and relations to accurately express the rights owned by the parties. ODRL has been adopted as a standard language by the Open Mobile Alliance (OMA) and OpenIPMP project. The core model [6] of ODRL 1.1 is shown in Fig.1. The data dictionary of ODRL 1.1 includes some entities: Rights, Asset, Party, Offer, Agreement, Revoke, Permission, Action, Prohibition, Constraint, Requirement, Condition, Encryption Digest / Key and Digital Signature, etc.

The use license often plays the role of an agreement between the user and the copyright owner. While in most Digital Rights Management (DRM) systems, the creation of a license has no involvement of consumers, but issued directly from the content provider. With digital content becoming more and more abundant, especially in the field of digital publishing, lots of new business models are emerging. For example, some publishers and readers may want to have more flexible fine-grained authorization, and they hope to update and combine the authorization during the distribution of e-books, digital newspapers and so on. There are some scenarios in detail as follows:

1) Content providers expect to provide the service in which readers can purchase a book by chapter and they can also have different privileges to the purchased chapters, such as reading, excerpting and printing. The scenario of fine-grained authorization is pictured as in Fig.2.

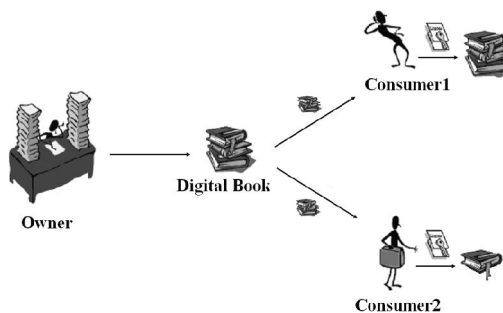


Figure 2. Scenario of fine-grained authorization

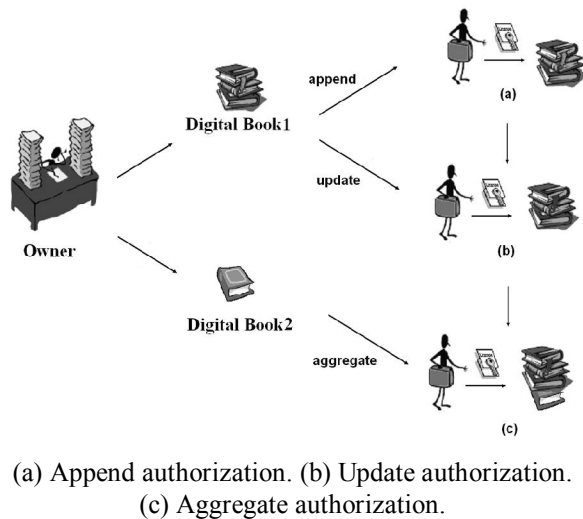


Figure 3. Scenario of dynamic authorization management.

2) After the readers have bought some chapters of a book, they can continue to purchase other chapters in the manner of progressive authorization or get the additional excerpting and printing privileges by updating their authorization. The scenarios of appending authorization and updating authorization are shown in Fig. 3-(a) and 3-(b) respectively.

3) Some readers want to have personalized services of re-combining the books, i.e., they would like some chapters of a book and chapters of another book to be packaged as a new book. Similarly some users may need the same channels of several newspapers (such as sports game, etc.). This kind of combination authorization needs to merge the content and licenses which have been bought by the readers, which is shown in Fig. 3-(c).

ODRL 1.1 can support fine-grained authorization, but did not provide the model and syntax about license appending, updating, and license aggregating. In this paper, we proposed an extended rights expression model to support the dynamic digital rights management such that a fine-grained content protection and convenient license management have been implemented to meet the needs of new digital publishing market and copyright protection.

The rest of this paper is organized as follows: Section II provides an overview of related work. In section III, we address the extended model in detail. Section IV introduces our dynamic rights management examples. Section V presents our model analysis and discussion. Finally, section VI concludes the paper.

II. Related Work

With the development of DRM systems, consumers want to get more personalized experience when purchasing digital content. So authorization on demand became a focus of study by utilizing the license negotiation mechanism.

In 2004, Jamkhedkar et al. [7] proposed a layered DRM system model, in which it is possible for a copyright holder to negotiate with a user in the negotiation layer. In 2006, Mikko Löytynoja et al. [8] introduced a License Negotiation System

(LNS) in the mobile DRM platform. The LNS can find all prospective licenses from system directories of the content

Copyright Owner

User

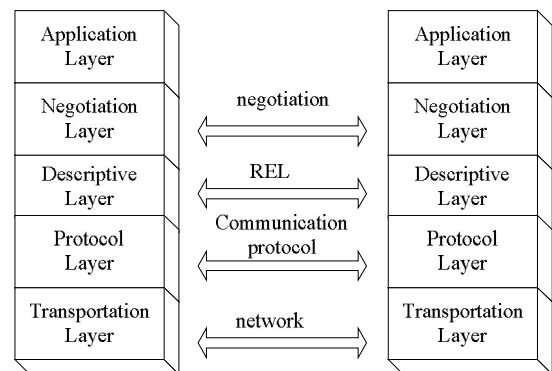


Figure 4. Layered model of license negotiation

provider according to the user's demand, and then the user can select his favorite license or reject all the listed options. In 2007, Arnab et al. [9] presented a formal model of DRM to cater for negotiations and express access control policies in a use license. These conceptual models and systems attempted to provide license negotiation mechanism and authorization on demand services, but they did not refer to the management of license. In our proposed model, we not only discuss license negotiation, but also present a license management scheme to satisfy the new need of e-book industry.

There are also some works about license negotiation by using Rights Expression Language (REL). First of all, let us preview the role of REL in expressing license negotiation. The layered model of license negotiation is shown in Fig. 4. We can see that REL can describe the user's authorization, and the rights implementation can be completed by communication protocols and data transportation. However, if we directly use the REL to express negotiation request and then negotiation result is converted into a license, there is no need to transform between communication protocols and REL. Therefore, rights expression language should provide a description of interactive mechanism and license negotiation to avoid the ambiguous transformation and system costs between REL and authorization protocols.

Since the traditional REL, such as Xrml and ODRL, lacks the description of negotiation and interaction, Arnab et al. [10-12] extended ODRL and XrML to support bi-directional communication between the copyright owner and a user. In [10], Arnab et al. extended ODRL to support request from a user and the owner can grant or deny the request. Our proposed model is similar with this scheme, but the main RequestType model is different. Especially, our model supports personalized combination authorization and dynamic license management of digital content.

In 2005, Arnab et al. [11] discussed the use of license negotiation in DRM systems to support fair use, but the scheme only deals with the negotiation process and has no dynamic license management such as appending, updating and aggregating an authorized license.

In [12], Arnab et al. detailed the license negotiation protocols by utilizing the draft of ODRL 2.0. An obvious difference between our model and their protocols in license

negotiation is that we particularly discussed fine-gained authorization. Moreover, license management is also covered in our proposed model. The latest draft of ODRL 2.0 [13] has defined a request policy to enable a consumer to propose usage terms to the copyright owner. However, ODRL 2.0 does not specify the action of progressive authorization and combination authorization. As is well known, ODRL 2.0 is not a stable release version. Thus we extended ODRL 1.1 to implement the license negotiation and license management. But it is noted that our model can also be implemented with the latest draft of ODRL 2.0.

ODRL 1.1 is able to express fine-grained authorization of digital content by setting the Constraint entity of rights. For example, if a user is interested in some chapters or pages of an e-book, ODRL can set authorization policies to limit the user’s access to chapters or pages. ODRL also can limit the surrounding reading, excerpting and printing privileges of chapters. However, ODRL 1.1 model lacks the description of negotiation and interaction. In addition, it has no description about dynamic license management which is very important in digital publishing.

Dongyang Xu et al. [14] proposed an extended rights expression model supporting dynamic authorization management in DRM systems. This paper is a detailed version which extends the ODRL 1.1 model to completely support fine-grained authorization, progressive or updating authorization, and combination authorization to achieve dynamic rights management of digital content. Without loss of generality, we mainly discuss our examples and implementation in e-book scenarios.

III. Extended Model Supporting License Management

ODRL 1.1 model can support three types of rights, i.e., offer, agreement and revoke. The copyright owner can declare some rights, issue or revoke these rights for digital content. However, ODRL 1.1 model does not define the rights request from a consumer, which can not meet the interactive license management needs. Thus we extended the ODRL rightsType model with adding a request element, as is shown in Fig. 5. A consumer can trigger a license management request to the copyright owner, such as appendPolicy request, updatePolicy request and aggregatePolicy request. Fig. 6 shows these three types of dynamic license management request. The extended requestType model must contain a party item with the role of assignee who initiates the request.

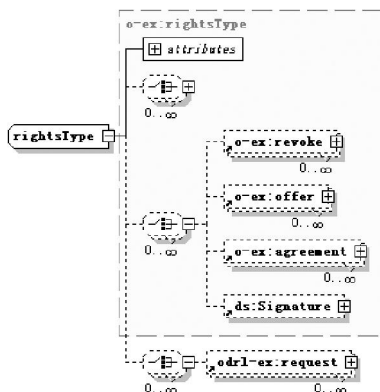


Figure 5. Extended rightsType model

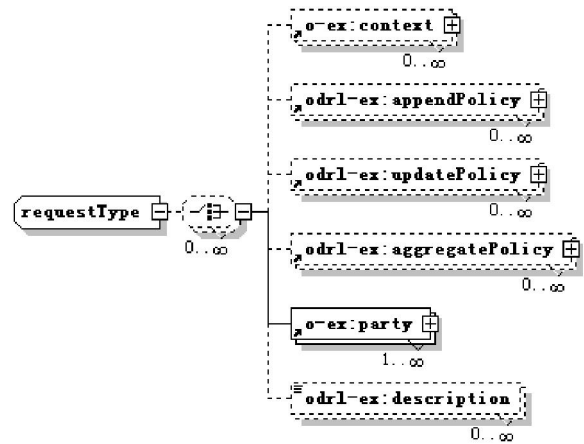


Figure 6. RequestType model

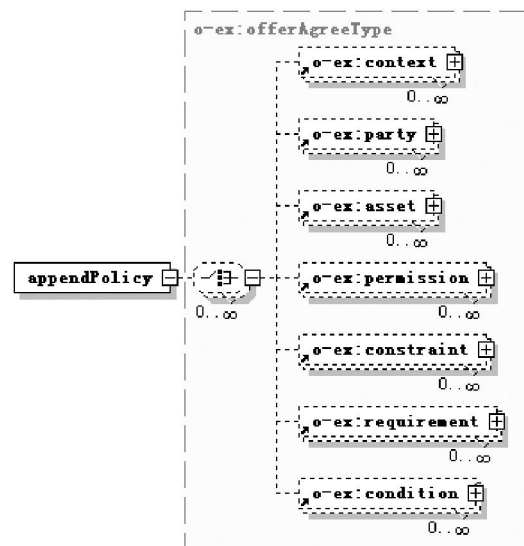


Figure 7. AppendPolicy model

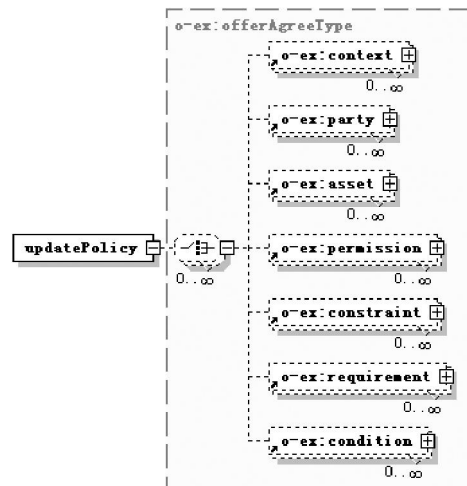


Figure 8. UpdatePolicy model

The appendPolicy request model is shown in Fig. 7. The new appendPolicy element is a data type of offerAgreeType

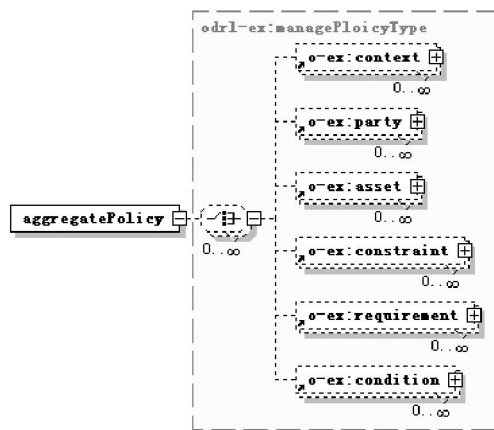


Figure 9. AggregatePolicy model

which has already been defined by ODRL 1.1. The appendPolicy model can express progressive authorization to a consumer. Similarly, we defined a new updatePolicy element to express updating authorization. Fig. 8 shows the updatePolicy request model. The updatePolicy element is also a data type of offerAgreeType. When a user aggregates authorization, there is no need to add new privileges. The user just describes the license identities he wants to combine in the aggregatePolicy request and initiates the request. Then the copyright owner can verify the validity of these license identities. If the license identities are valid and have been authorized to the user, then the copyright owner aggregates all the licenses; otherwise, it rejects the user's request. Thus, we defined a managePolicyType data type which excludes Permission element. The new aggregatePolicy element is a data type of managePolicyType, as is shown in Fig. 9.

The above extended ODRL model can easily support the interactive mechanism and license negotiation to dynamically manage the license. A user can initiate a license management request of digital content and the license server responds with the request. If the user's request is valid, the license server can append, update or aggregate the license of the content according to the request, otherwise reject the user's request. The digital rights expression model supporting dynamic license management can solve the problem of owning many fragment licenses of the same digital content. It also provides a solution for personalized combination authorization of many segments from lots of different digital content. The combination authorization can satisfy the need of a consumer who selects different chapters from many favorite books using the process of re-combination for convenience.

IV. Example of Extended ODRL Model

A. Fine-grained authorization

Some users may be interested in some chapters or pages of an e-book, and purchase some privileges of the book, such as reading, printing, and excerpting. As a result, it is necessary to issue the fine-grained authorization of digital content. Fine-grained authorization of digital content is compatible with the whole authorization, that is, the user can either purchase an entire book including all privileges or buy parts of the book and privileges. Fine-grained authorization is a

basis of dynamic rights management and upholds the progressive authorization, updating authorization and combination authorization of digital content. ODRL 1.1 can provide fine-grained authorization of digital content by setting the Constraint element of Permission entity, as shown in e-book example 1 and Fig. 10 ~ Fig. 12. For convenience of description in following e-book examples, we omit the XML header information, encryption key and signature information of content.

E-book example 1:

A bookstore retailer named Distributor1 released a list of books for sale, including the book named Book1 whose title is "Introducing AutoCAD". Distributor1 defined a price strategy for Book1. The price strategy is that the whole book is for 20 CNY, each chapter for 2 CNY and outline or introduction of the book can be previewed for free. A user named Consumer1 is interested in Book1 and pays 20 CNY for a full content, while another user named Consumer2 previews the book's introduction and decides to pay 2 CNY for the second chapter of the book.

```
<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex=http://odrl.net/1.1/ODRL-EX
xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
xmlns:onix="http://www.editeur.org/onix/ReferenceNames">
  <o-ex:offer>
    <o-ex:asset> <o-ex:context>
      <o-dd:uid>http://example.com/book1</o-dd:uid>
      <o-dd:name> Introducing AutoCAD </o-dd:name>
    </o-ex:context> </o-ex:asset>
    <o-ex:permission>
      <o-dd:display><o-ex:constraint>
        <o-dd:unit o-ex:type="onix:NumberOfPages">
          <o-ex:constraint><o-dd:range>
            <o-dd:min>1</o-dd:min> <o-dd:max>7</o-dd:max>
          </o-dd:range> </o-ex:constraint>
        </o-dd:unit>
      </o-ex:constraint></o-dd:display>
    </o-ex:permission>
    <o-ex:permission>
      <o-dd:display><o-ex:constraint></o-dd:display>
      <o-ex:requirement>
        <o-dd:prepay><o-dd:payment>
          <o-dd:amount o-dd:currency="CNY">20.00</o-dd:amount>
        </o-dd:payment></o-dd:prepay>
      </o-ex:requirement> </o-ex:permission>
    <o-ex:permission>
      <o-dd:display><o-ex:constraint>
        <o-dd:unit o-ex:type="onix:NumberOfPages">
          <o-ex:constraint> <o-dd:range>
            <o-dd:min>8</o-dd:min> <o-dd:max>36</o-dd:max>
          </o-dd:range> </o-ex:constraint>
        </o-dd:unit>
      </o-ex:constraint></o-dd:display>
      <o-ex:requirement>
        <o-dd:prepay><o-dd:payment>
          <o-dd:amount o-dd:currency="CNY">2.00</o-dd:amount>
        </o-dd:payment> </o-dd:prepay>
      </o-ex:requirement> </o-ex:permission>
    .....
    <o-ex:party>
      <o-ex:context> <o-dd:uid>http://example.com/pup</o-dd:uid>
      </o-ex:context> </o-ex:party>
  </o-ex:offer></o-ex:rights>
```

Figure 10. Example of a price strategy for Book1

```
<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
  <o-ex:agreement>
    <o-ex:context><o-dd:uid>http://example.com/policy:001
    </o-dd:uid></o-ex:context>
    <o-ex:asset><o-ex:context>
      <o-dd:uid>http://example.com/book1</o-dd:uid>
      <o-dd:name>Introducing AutoCAD</o-dd:name>
    </o-ex:context> </o-ex:asset>
    <o-ex:permission>
      <o-dd:display><o-ex:constraint></o-dd:display>
      <o-ex:requirement>
        <o-dd:prepay><o-dd:payment>
          <o-dd:amount o-dd:currency="CNY">20.00
          </o-dd:amount> </o-dd:payment> </o-dd:prepay>
        </o-ex:requirement>
      </o-ex:permission>
      <o-ex:party><o-ex:context>
        <o-dd:uid>http://example.com/pup</o-dd:uid>
      </o-ex:context></o-ex:party>
      <o-ex:party><o-ex:context>
        <o-dd:uid>http://example.com/Consumer1</o-dd:uid>
      </o-ex:context></o-ex:party>
    </o-ex:agreement>
  </o-ex:rights>
```

Figure 11. Example of whole authorization license

```
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
  <o-ex:agreement><o-ex:context>
    <o-dd:uid>http://example.com/policy:002</o-dd:uid>
  </o-ex:context>
  <o-ex:asset><o-ex:context>
    <o-dd:uid>http://example.com/book1</o-dd:uid>
    <o-dd:name>Introducing AutoCAD</o-dd:name>
  </o-ex:context> </o-ex:asset>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint>
      <o-dd:unit o-ex:type="onix:NumberOfPages">
      <o-ex:constraint><o-dd:range> <o-dd:min>1</o-dd:min>
      <o-dd:max>7</o-dd:max></o-dd:range></o-ex:constraint>
    </o-dd:unit></o-ex:constraint></o-dd:display>
  </o-ex:permission>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint>
      <o-dd:unit o-ex:type="onix:NumberOfPages">
      <o-ex:constraint><o-dd:range>
        <o-dd:min>37</o-dd:min>
        <o-dd:max>81</o-dd:max>
      </o-dd:range></o-ex:constraint>
    </o-dd:unit></o-ex:constraint></o-dd:display>
    <o-ex:requirement>
      <o-dd:prepay><o-dd:payment>
        <o-dd:amount o-dd:currency="CNY">2.00</o-dd:amount>
        </o-dd:payment></o-dd:prepay>
      </o-ex:requirement>
    </o-ex:permission>
    <o-ex:party><o-ex:context>
      <o-dd:uid>http://example.com/pup</o-dd:uid>
    </o-ex:context></o-ex:party>
    <o-ex:party><o-ex:context>
      <o-dd:uid>http://example.com/Consumer2</o-dd:uid>
    </o-ex:context>
  </o-ex:party>
</o-ex:agreement >
</o-ex:rights>
```

Figure 12. Example of fine-grained authorization license

B. Progressive and updating authorization

Authorized users can continue to buy the other chapters of the book with progressive authorization through the appendPolicy request after they have bought some chapters with fine-grained authorization. On the other hand, the authorized users can also initiate an updatePolicy request to obtain other privileges they have not owned, such as excerpting or printing permission of a book. E-book Example 2 and Example 3 are the progressive authorization and updating authorization respectively. The detailed description of extended <odrl-ex:request>, <odrl-ex:appendPolicy>, <odrl-ex:updatePolicy> is shown in Fig. 13 and Fig. 15.

E-book Example 2:

A bookstore retailer named Distributor1 released a list of books for sale. A user named Consumer2 have purchased the second chapter of a book named "Introducing AutoCAD". A month later, Consumer2 again paid 2 CNY to Distributor1 for the third chapter of the book.

An example of appendPolicy request is shown in Fig.13. If the license server grants the request, it will generate a new progressive authorization license which is described in Fig. 14.

E-book Example 3:

The price strategy of a book named "Introducing AutoCAD" is that the whole book is for 20 CNY, reading privilege per chapter for 2 CNY and printing privilege per chapter for 1 CNY. A user named Consumer2 have purchased the second chapter of this book with only a reading privilege. A week later, Consumer2 thought the chapter was very interesting, and decided to get the printing permission of the second chapter. So Consumer2 paid 1 CNY and initiated an updating authorization request to obtain the printing and reading permissions of the second chapter.

```
<o-ex:rights>
  <odrl-ex:request>
    <o-ex:context><o-dd:uid>http://example.com/policy:003
    </o-dd:uid></o-ex:context>
    <odrl-ex:appendPolicy>
      <o-ex:asset><o-ex:context>
        <o-dd:uid>http://example.com/book1</o-dd:uid>
        <o-dd:name>Introducing AutoCAD</o-dd:name>
      </o-ex:context> </o-ex:asset>
      <o-ex:permission>
        <o-dd:display>
          <o-ex:constraint><o-dd:unit
            o-ex:type="onix:NumberOfPages">
          <o-ex:constraint><o-dd:range>
            <o-dd:min>82</o-dd:min><o-dd:max>101</o-dd:max>
          </o-dd:range></o-ex:constraint>
          </o-dd:unit></o-ex:constraint>
        </o-dd:display>
      </o-ex:permission>
      <o-ex:party><o-ex:context>
        <o-dd:uid>http://example.com/Consumer2</o-dd:uid>
      </o-ex:context></o-ex:party>
    </odrl-ex:appendPolicy>
  </odrl-ex:request>
</o-ex:rights>
```

Figure 13. Example of appendPolicy request

```

<o-ex:rights>
  <o-ex:agreement><o-ex:context>
    <o-dd:uid>http://example.com/policy:004</o-dd:uid>
  </o-ex:context>
  <o-ex:asset><o-ex:context>
    <o-dd:uid>http://example.com/book1</o-dd:uid>
    <o-dd:name>Introducing AutoCAD </o-dd:name>
  </o-ex:context> </o-ex:asset>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint>
      <o-dd:unit o-ex:type="onix:NumberOfPages">
        <o-ex:constraint><o-dd:range>
          <o-dd:min>1</o-dd:min><o-dd:max>7</o-dd:max>
        </o-dd:range></o-ex:constraint>
      </o-dd:unit></o-ex:constraint></o-dd:display>
    </o-ex:permission>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint>
      <o-dd:unit o-ex:type="onix:NumberOfPages">
        <o-ex:constraint><o-dd:range>
          <o-dd:min>37</o-dd:min>
          <o-dd:max>81</o-dd:max>
        </o-dd:range></o-ex:constraint>
      </o-dd:unit></o-ex:constraint></o-dd:display>
    <o-ex:requirement>
      <o-dd:prepay><o-dd:payment><o-dd:amount
        o-dd:currency="CNY">2.00</o-dd:amount>
      </o-dd:payment></o-dd:prepay>
    </o-ex:requirement>
  </o-ex:permission>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint><o-dd:unit
      o-ex:type="onix:NumberOfPages">
        <o-ex:constraint><o-dd:range>
          <o-dd:min>82</o-dd:min>
          <o-dd:max>101</o-dd:max>
        </o-dd:range></o-ex:constraint>
      </o-dd:unit></o-ex:constraint></o-dd:display>
    <o-ex:requirement>
      <o-dd:prepay><o-dd:payment><o-dd:amount
        o-dd:currency="CNY">2.00</o-dd:amount>
      </o-dd:payment></o-dd:prepay>
    </o-ex:requirement>
  </o-ex:permission>
  <o-ex:party><o-ex:context>
    <o-dd:uid>http://example.com/pup</o-dd:uid>
  </o-ex:context></o-ex:party>
  <o-ex:party><o-ex:context>
    <o-dd:uid>http://example.com/Consumer2</o-dd:uid>
  </o-ex:context></o-ex:party></o-ex:agreement >
</o-ex:rights>

```

Figure 14. Example of progressive authorization license

An example of updatePolicy request is shown in Fig.15. The consumer requests a print privilege of the second chapter from book1. If the license server grants the request, it will generate a new updating authorization license which is described in Fig. 16.

The progressive authorization or updating authorization enables the users to achieve fine-grained authorization while they do not need to maintain multiple fragment licenses of the same digital content. This flexible license management property can meet the special requirements of personalized users. In addition, the efficient license management increases user's satisfaction towards digital rights management technologies and promotes the distribution of legal digital content.

```

<o-ex:rights>
  <odrl-ex:request>
    <o-ex:context>
      <o-dd:uid>http://example.com/policy:005</o-dd:uid>
    </o-ex:context>
  <odrl-ex:updatePolicy>
    <o-ex:asset>
      <o-ex:context>
        <o-dd:uid>http://example.com/book1</o-dd:uid>
        <o-dd:name>Introducing AutoCAD</o-dd:name>
      </o-ex:context>
    </o-ex:asset>
    <o-ex:permission>
      <o-dd:print><o-ex:constraint>
        <o-dd:unit o-ex:type="onix:NumberOfPages">
          <o-ex:constraint><o-dd:range>
            <o-dd:min>37</o-dd:min>
            <o-dd:max>81</o-dd:max>
          </o-dd:range></o-ex:constraint>
        </o-dd:unit></o-ex:constraint>
      </o-dd:print>
    </o-ex:permission>
  </o-ex:party>
    <o-ex:context>
      <o-dd:uid>http://example.com/Consumer2</o-dd:uid>
    </o-ex:context>
  </o-ex:party>
</odrl-ex:updatePolicy>
</odrl-ex:request>
</o-ex:rights>

```

Figure 15. Example of updatePolicy request

```

<o-ex:rights>
  <o-ex:agreement><o-ex:context>
    <o-dd:uid>http://example.com/policy:006</o-dd:uid>
  </o-ex:context>
  <o-ex:asset><o-ex:context>
    <o-dd:uid>http://example.com/book1</o-dd:uid>
    <o-dd:name> Introducing AutoCAD </o-dd:name>
  </o-ex:context> </o-ex:asset>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint><o-dd:unit
      o-ex:type="onix:NumberOfPages">
        <o-ex:constraint><o-dd:range>
          <o-dd:min>1</o-dd:min><o-dd:max>7
        </o-dd:max></o-dd:range></o-ex:constraint>
      </o-dd:unit></o-ex:constraint></o-dd:display>
    </o-ex:permission>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint><o-dd:unit
      o-ex:type="onix:NumberOfPages">
        <o-ex:constraint><o-dd:range>
          <o-dd:min>37</o-dd:min>
          <o-dd:max>81</o-dd:max>
        </o-dd:range></o-ex:constraint>
      </o-dd:unit></o-ex:constraint></o-dd:display>
    <o-dd:print ><o-ex:constraint><o-dd:unit
      o-ex:type="onix:NumberOfPages">
        <o-ex:constraint><o-dd:range>
          <o-dd:min>37</o-dd:min><o-dd:max>81
        </o-dd:max></o-dd:range></o-ex:constraint>
      </o-dd:unit></o-ex:constraint></o-dd:print >
    </o-ex:requirement>
    <o-dd:prepay><o-dd:payment><o-dd:amount
      o-dd:currency="CNY">1.00</o-dd:amount>
    </o-dd:payment></o-dd:prepay>
  </o-ex:requirement>
  </o-ex:permission>
  <o-ex:party><o-ex:context>
    <o-dd:uid>http://example.com/pup</o-dd:uid>
  </o-ex:context></o-ex:party>
  <o-ex:party><o-ex:context>
    <o-dd:uid>http://example.com/Consumer2</o-dd:uid>
  </o-ex:context></o-ex:party></o-ex:agreement >
</o-ex:rights>

```

Figure 16. Example of updating authorization license

C. Combination authorization

Combination authorization can implement personalized privileges and control through re-combination of multiple segments of different digital content, which is very convenient for users to effectively manage licenses. Combination authorization is a new business model. It can reduce the cost of a user to buy the digital content, and greatly meet the personalized needs of the user. On the other hand, the copyright owner can also provide more flexible authorization to consumers. Hence, combination authorization can promote the purchasing and dissemination of digital content.

Combination authorization can be implemented through our proposed aggregatePolicy request model. The process is that a user initiates an aggregatePolicy request and the license server verifies the relevant issued licenses. If the verification is passed, the license server will execute an aggregating operation and make the combined digital content and license ready for the user to download. E-book Example 4 shows the combination authorization.

```

<o-ex:rights>
  <o-ex:agreement><o-ex:context>
    <o-dd:uid>http://example.com/policy:007</o-dd:uid>
  </o-ex:context>
  <o-ex:asset><o-ex:context>
    <o-dd:uid>http://example.com/book2</o-dd:uid>
    <o-dd:name> Architectural Drawing Tutorial Using
      AutoCAD </o-dd:name>
  </o-ex:context> </o-ex:asset>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint><o-dd:unit
      o-ex:type="onix:NumberOfPages">
    <o-ex:constraint><o-dd:range>
      <o-dd:min>186</o-dd:min>
      <o-dd:max>192</o-dd:max>
    </o-dd:range></o-ex:constraint>
    </o-dd:unit></o-ex:constraint></o-dd:display>
  <o-ex:requirement>
    <o-dd:prepay><o-dd:payment><o-dd:amount
      o-dd:currency="CNY">1.00</o-dd:amount>
    </o-dd:payment></o-dd:prepay>
  </o-ex:requirement>
</o-ex:permission>
  <o-ex:permission>
    <o-dd:display><o-ex:constraint><o-dd:unit
      o-ex:type="onix:NumberOfPages">
    <o-ex:constraint><o-dd:range>
      <o-dd:min>193</o-dd:min>
      <o-dd:max>202</o-dd:max>
    </o-dd:range></o-ex:constraint>
    </o-dd:unit></o-ex:constraint></o-dd:display>
  <o-ex:requirement>
    <o-dd:prepay><o-dd:payment><o-dd:amount
      o-dd:currency="CNY">1.00</o-dd:amount>
    </o-dd:payment></o-dd:prepay>
  </o-ex:requirement>
</o-ex:permission>
  <o-ex:party><o-ex:context>
    <o-dd:uid>http://example.com/pup</o-dd:uid>
  </o-ex:context></o-ex:party>
  <o-ex:party><o-ex:context>
    <o-dd:uid>http://example.com/Consumer2</o-dd:uid>
  </o-ex:context>
</o-ex:party>
</o-ex:agreement >
</o-ex:rights>
    
```

Figure 17. A license of another reference book

```

<o-ex:rights>
  <odrl-ex:request><o-ex:context>
    <o-dd:uid>http://example.com/policy:008</o-dd:uid>
  </o-ex:context>
  <odrl-ex:aggregatePolicy>
    <o-ex:asset><o-ex:context>
      <o-dd:uid>http://example.com/policy:002</o-dd:uid>
      <o-dd:name>Introducing AutoCAD</o-dd:name>
    </o-ex:context>
    </o-ex:asset>
    <o-ex:asset><o-ex:context>
      <o-dd:uid>http://example.com/policy:007</o-dd:uid>
      <o-dd:name>Architectural Drawing Tutorial Using
        AutoCAD</o-dd:name>
    </o-ex:context>
    </o-ex:asset>
  <o-ex:party>
    <o-ex:context>
      <o-dd:uid>http://example.com/Consumer2
    </o-dd:uid>
    </o-ex:context>
  </o-ex:party>
</odrl-ex: aggregatePolicy>
</odrl-ex:request>
</o-ex:rights>
    
```

Figure 18. Example of aggregatePolicy request

E-book example 4:

A Teacher taught the AutoCAD course and specified some chapters of reference books. A student named Consumer2 has bought the second and third chapters of a reference book titled "Introducing AutoCAD". Consumer2 also purchased Chapter 8 and Chapter 9 of the specified reference book "Architectural Drawing Tutorial Using AutoCAD". Consumer2 wanted a personalized service to combine these chapters and licenses of the two books for easy reading and reviewing.

Fig. 17 illustrates the license of another reference book which has been bought by Consumer2. The detailed description of extended XML element <odrl-ex:request>, <odrl-ex:aggregatePolicy> is shown in Fig. 18 and Fig. 19 shows a example license of combination authorization.

Combination authorization can implement convenient maintenance and management of multiple fragment licenses. The objects of combination authorization are not only from the same electronic document, but also can be from multiple segments of different documents.

Combination authorization is not just simply merging multiple licenses. Simply merging licenses has the following deficiencies: 1) it can not accord with the XML Schema of license specification. 2) A lot of duplicate information of permission may be introduced. These duplicate information leads to lower efficiency when parsing the XML file. 3) There exists a signature of license server in the original license file. If we simply remove the redundant XML node or modify the content of the XML file, the signature verification process will be failed. In addition, during combination authorization the license server needs to re-package the digital content in accordance with the segment-based document format if the

objects are from multiple source document segments. Then, the license server generates the new combination authorization license whose specification is listed in Fig. 19.

```

<o-ex:rights>
  <o-ex:agreement><o-ex:context>
    <o-dd:uid>http://example.com/policy:009</o-dd:uid>
  </o-ex:context>
  <o-ex:asset id="ASSET-01">
    <o-ex:context>
      <o-dd:uid>http://example.com/book1</o-dd:uid>
      <o-dd:name>Introducing AutoCAD</o-dd:name>
    </o-ex:context></o-ex:asset>
  <o-ex:asset id="ASSET-02">
    <o-ex:context>
      <o-dd:uid>http://example.com/book2</o-dd:uid>
      <o-dd:name>
        Architectural Drawing Tutorial Using AutoCAD
      </o-dd:name>
    </o-ex:context>
  </o-ex:asset>
  <o-ex:permission>
    <asset idref="ASSET-01">
      <o-dd:display><o-ex:constraint>
        <o-dd:unit o-ex:type="onix:NumberOfPages">
          <o-ex:constraint>
            <o-dd:range><o-dd:min>1</o-dd:min>
            <o-dd:max>7</o-dd:max></o-dd:range>
          </o-ex:constraint></o-dd:unit></o-ex:constraint>
        </o-dd:display>
      </o-ex:permission>
    </o-ex:permission>
    <asset idref="ASSET-01">
      <o-dd:display><o-ex:constraint>
        <o-dd:unit o-ex:type="onix:NumberOfPages">
          <o-ex:constraint>
            <o-dd:range><o-dd:min>37</o-dd:min>
            <o-dd:max>81</o-dd:max></o-dd:range>
          </o-ex:constraint></o-dd:unit></o-ex:constraint>
        </o-dd:display>
      <o-ex:requirement><o-dd:prepay><o-dd:payment>
        <o-dd:amount o-dd:currency="CNY">2.00
        </o-dd:amount></o-dd:payment></o-dd:prepay>
      </o-ex:requirement>
    </o-ex:permission>
    .....
  <o-ex:permission>
    <asset idref="ASSET-02">
      <o-dd:display><o-ex:constraint>
        <o-dd:unit o-ex:type="onix:NumberOfPages">
          <o-ex:constraint>
            <o-dd:range><o-dd:min>186</o-dd:min>
            <o-dd:max>192</o-dd:max></o-dd:range>
          </o-ex:constraint></o-dd:unit></o-ex:constraint>
        </o-dd:display>
      <o-ex:requirement><o-dd:prepay><o-dd:payment>
        <o-dd:amount o-dd:currency="CNY">1.00
        </o-dd:amount></o-dd:payment></o-dd:prepay>
      </o-ex:requirement>
    </o-ex:permission>
    .....
  <o-ex:party>
    <o-ex:context>
      <o-dd:uid>http://example.com/pup</o-dd:uid>
    </o-ex:context>
  </o-ex:party>
  <o-ex:party>
    <o-ex:context>
      <o-dd:uid>http://example.com/Consumer2</o-dd:uid>
    </o-ex:context>
  </o-ex:party>
</o-ex:agreement>
</o-ex:rights>

```

Figure 19. Example of combination authorization license

V. Discussions

A. Model analysis

We have proposed an extended ODRL model, including the extension of rightsType model by adding a request action and the new appendPolicy, updatePolicy, aggregatePolicy model. These extended models can describe the license management request and support fine-grained authorization, progressive authorization, updating authorization and combination authorization. It is noteworthy that if a user wants to delete authorization of digital content, he just needs to delete his local license and is not required interacting with the license server. Directly using the REL to describe the license negotiation protocols can avoid the ambiguous transformation between REL and authorization protocols and enhance the efficiency of license management.

B. Flexibility and personalization

The extended ODRL model we proposed is flexible during the process of authorization and supports personalized authorization on demand. Fine-grained authorization provides a more flexible business model because a book can be sold by chapter or by privilege, such as reading, excerpting or printing. Promotion of some chapters or free preview can also be realized by fine-grained authorization. In addition, the progressive authorization or updating authorization alleviates the user to maintain multiple chapter licenses of the same e-book. Consumers can also obtain personalized services with combination authorization to re-combine multiple segments of different digital content. The personalized services are very convenient for users to effectively manage and store the correlative books and licenses.

VI. Conclusion

In this paper, we have extended ODRL 1.1 rights expression model to support dynamic digital rights management. We also listed some examples of the extended model. In a word, the extended model has the following properties:

- 1) The extended model can describe the license negotiation protocols to avoid the ambiguous transformation between REL and authorization protocols.
- 2) It supports fine-grained authorization, progressive authorization, updating authorization and combination authorization to achieve dynamic rights management of digital content.
- 3) It is flexible during the process of authorization and supports personalized authorization on demand.

Our model provides a practical solution to the license negotiation and license management of digital content. The future work includes describing dynamic rights management using other REL and a formal language to define the authorization policy.

Acknowledgment

This work was supported by Major Scientific and Technological Project of Press and Publication of China (No. GXTC-CZ-1015004/02).

References

- [1] W. Rosenblatt, W. Trippe and S. Mooney. *Digital Rights Management: Business and Technology*, M&T Books, New York, 2002.
- [2] Y. Yu and Z. Tang. "A survey of the research on digital rights management", *Chinese Journal of Computers*, vol. 28, no. 12, pp. 1957-1968, 2005.
- [3] W. Zeng, H. Yu, C. Lin. *Multimedia Security Technologies for Digital Rights Management*, Academic Press, 2006.
- [4] C. N. Chong, R. Corin, S. Etalle, P. H. Hartel, W. Jonker, and Y. W. Law. "LicenseScript: A novel digital rights language and its semantics", In *Third International Conference on the Web Delivery of Music*, pp. 122-129, Los Alamitos, USA, 2003.
- [5] Contentguard. *XrML 2.0 Technical Overview*, <http://www.xrml.org/Reference/XrMLTechnicalOverviewwV1.pdf>, 2002.
- [6] R. Iannella. *Open Digital Rights Language (ODRL) 1.1*, IPR Systems Pty Ltd, URL: <http://odrl.net/1.1/ODRL-11.pdf>, August, 2002.
- [7] P. A. Jamkhedkar, G. L. Heileman. "DRM as a Layered System", *ACM Digital Rights Management workshop*, Washington DC, USA, 2004.
- [8] M. Löytynoja, T. Koskela, M. Brockman, T. Seppänen. "Mobile DRM-enabled Multimedia Platform for Peer-to-Peer Applications", *Proc. of the Eighth IEEE International Symposium on Multimedia ISM'06*, San Diego, USA, 2006.
- [9] Alapan Arnab, Andrew Hutchison. "Persistent Access Control: A Formal Model for DRM", *Proceedings of the ACM Digital Rights Management workshop DRM'07*, 2007.
- [10] A. Arnab, A. Hutchison. "Extending ODRL to Enable Bi-Directional Communication", In *Proceedings of the 2nd International ODRL Workshop*, Lisbon, Portugal, 2005.
- [11] A. Arnab, A. Hutchison. "Fairer usage contracts for DRM", In *Proceedings of the fifth ACM Workshop on Digital Rights Management*, Alexandria, USA, 2005.
- [12] A. Arnab, A. Hutchison. "DRM Use License Negotiation using ODRL v2.0", In *Proceedings 5th International Workshop for Technology, Economy, and Legal Aspects of Virtual Goods, incorporating the 3rd International ODRL Workshop*, Koblenz, Germany, 2007.
- [13] R. Iannella, S. Guth, H. Hundacker, D. Paehler and A. Kasten. *Open Digital Rights Language (ODRL) Version 2.0 - Core Model*, Draft Specification, ODRL Initiative, URL: <http://odrl.net/2.0/DS-ODRL-Model.html>, 17 February 2011.
- [14] Dongyang Xu, Zhi Tang, Yinyan Yu. "An Extended Rights Expression Model Supporting Dynamic Digital Rights Management", In *Proceedings of the Seventh International Conference on Information Assurance and Security*, pp. 186-191, 2011.

Appendix

In this section, we provide the listing of the extended ODRL schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:odrl-ex="http://www.icst.pku.edu.cn/SDDRM-ODRL"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  targetNamespace="http://www.icst.pku.edu.cn/SDDRM-ODRL"
  elementFormDefault="qualified"
  attributeFormDefault="qualified"
  version="0.5">
```

```
<xs:import namespace="http://odrl.net/1.1/ODRL-EX"
  schemaLocation="http://www.odrl.net/1.1/ODRL-EX-11.xsd"/>
<xs:annotation>
  <xs:documentation>XML Schema extends ODRL to
  manage the License. Created by Dongyang Xu
</xs:documentation>
</xs:annotation>
<xs:element name="rights" type="odrl-ex:rightsType"/>
<xs:element name="request"
  type="odrl-ex:requestType"/>
<xs:complexType name="requestType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="o-ex:context" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element ref="odrl-ex:appendPolicy"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="odrl-ex:updatePolicy"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="odrl-ex:aggregatePolicy"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="o-ex:party"
      maxOccurs="unbounded"/>
    <xs:element name="description" type="xs:string"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>
<xs:element name="appendPolicy"
  type="o-ex:offerAgreeType"/>
<xs:element name="updatePolicy"
  type="o-ex:offerAgreeType"/>
<xs:element name="aggregatePolicy"
  type="odrl-ex:managePloicyType"/>
<xs:complexType name="managePloicyType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="o-ex:context" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element ref="o-ex:party" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element ref="o-ex:asset" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element ref="o-ex:constraint" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element ref="o-ex:requirement" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element ref="o-ex:condition" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="rightsType">
  <xs:complexContent>
    <xs:extension base="o-ex:rightsType">
      <xs:choice minOccurs="0"
        maxOccurs="unbounded">
        <xs:element ref="odrl-ex:request"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:schema>
```

Author Biographies



Dongyang Xu is currently a Ph.D. Student in the Institute of Computer Science and Technology, Peking University, China. Prior he was a Computer Engineer at the Beijing Certificate Authority Corporation, and developed authentication and authorization related software. His current research focuses on digital rights management, access control, internet security, cryptography, and attribute-based encryption in cloud servers.



Zhi Tang was born in Zhejiang, China in 1965. He received his B.S. degree in radio physics from Peking University in 1987, and obtained his M.S. degree and Ph.D. degree in computer applied technology from Peking University in 1990 and 1995 respectively. He is a professor and doctoral supervisor in Institute of Computer Science and Technology of Peking University, China. His research interests include document processing and digital rights management.



Yinyan Yu received her B.S. degree in computational mathematics from Nanjing University of Science & Technology in 1998, M.S. degree in applied mathematics from Chinese Academy of Sciences in 2001, and Ph. D. degree in computer applied technology from Peking University in 2005. She is a senior engineer in Institute of Computer Science and Technology of Peking University, China. Her research interests include digital rights management and information security.



Cheng Qu received his B.S. degree in computer science and technology from Nankai University, China, in 2011. He is currently working toward the M.S. degree in the Institute of Computer Science and Technology, Peking University, China. His current research interests include dynamic rights management, rights expression language and access control.