A Review of the OMA DRM V2 ODRL Profile

Renato Iannella, *National ICT Australia (NICTA), Australia <renato@nicta.com.au>*

Abstract: This paper presents a review of the OMA DRM Version 2.0 profile of the ODRL REL. It looks at the decisions made by the OMA DRM working group and offers alternatives. The lessons are important to both the ODRL Initiative and to other groups developing profiles of the ODRL REL.

Index terms - ODRL, DRM Open Mobile Alliance, REL Profile

I. INTRODUCTION

The OMA DRM Version 2.0 specification [3] extended the profile adopted in the OMA DRM Version 1.0 specification [2]. The extensions included new elements - specific to the OMA community - and reuse of some of the standard ODRL data dictionary elements. A summary is shown in the Table below.

Permissions	Constraints
Play	Count
Display	Timed-Count *(OMA extension)*
Execute	Datetime
Print	Interval
Export *(OMA extension)*	Accumulated
	Individual
	System *(OMA Extension)*

This paper will review some of these elements and analyse the different options and issues in creating XML profiles of ODRL. See [1] for a discussion of the use of XML in ODRL.

II. THE COUNT ELEMENT

The OMA DRM REL required a more refined version of the "count" constraint. The requirement was to allow for a period of time to elapse before decrementing the counter. The rationale being that, in come cases such as audio media, the act of playing is not registered until a "few seconds" into the track. This would allow consumers to stop playing the track without effecting their count constraint if they are within this small period of time.

OMA decided to create a new Constraint to capture this refined count constraint. Alternatively, they could have extended the existing Count constraint, and hence, kept some level of interoperability (and backward compatibility) with other ODRL implementations, including OMA DRM version 1.0 systems.

All constraints can have any attribute from any other XML namespace (as defined in the XML Schema). Hence, OMA DRM could have just defined an additional attribute in the schema profile, such as:

```
<xsd:attribute name="timer"
      type="xsd:positiveInteger"/>
```

and then used this with the standard ODRL count element, such as:

```
<o-dd:count oma-dd:timer="30"> 10
      </o-dd:count>
```

Another option could have been to utilise the standard "type" attribute that can appear on all constraint elements. You would then need to de-

fine the structure of the URI value for “type” such as a URN prefix (eg "oma:reduce-state:") followed by a positive integer of seconds. For example:

```
<o-dd:count o-ex:type="oma:reduce-state:30">
    10 </o-dd:count>
```

The introduction of the timer element could also cause some confusion with different permissions. The OMA DRM WG recognised this with the export and print permission and explicitly disallow its use. However, its use with Display is unclear, as display will render static content only. The timed-count permission should have been limited to only time-based media content (eg audio, video, games).

III. THE EXPORT ELEMENT

The Export permission allows users to convert the content to other formats, and is aimed at supporting future interoperability and maximising the applicability of content across platforms. There are two attribute “modes” defined that control the export:

- move - the content is moved from the original device to another device and deleted from the original
- copy - the content is copied from the original device to another device and kept on the original device

The semantics of the export are very similar to that of ODRL’s “move” and “duplicate” permissions. For example, the following two elements would be equivalent:

```
<oma:export mode="move">
<o-dd:move>
```

and:

```
<oma:export mode="copy">
<o-dd:duplicate>
```

The issue here would be the trade-off between the level of equivalence of these statements, and wider interoperability.

IV. THE SYSTEM ELEMENT

The System element is a Constraint that is used by the Export permission. It constrains the export operation to specific “systems”. This maybe useful in ensuring that exported content only moves to platforms that can support DRM, although that is not mandatory.

There are a number of existing ODRL elements that could be used here:

- cpu - any system with a cpu
- hardware - any generic hardware device
- software - any software dependencies

The question would be what types of "exports" are envisaged with OMA DRM 2.0 and how can they be controlled?

Likely candidates include:

- other physical mobile devices
- other physical desktop devices
- specific (DRM) platforms must be present

Also, this process may include "conversion" of the content and Rights Object to another platform (eg Real Helix, Microsoft Media).

So there maybe a need to control both aspects of the "export" - the type of device and the platform - together and individually.

For example, to limit to other hardware, the following may suffice:

```
<o-dd:duplicate>
  <o-ex:constraint>
    <o-dd:hardware>
      <o-ex:context>
        <o-dd:uid>oma:apple:ipod</o-dd:uid>
      </o-ex:context>
    </o-dd:hardware>
  <o-ex:constraint>
</o-dd:duplicate>
```

And to limit to platforms:

```
<o-dd:move>
  <o-ex:constraint>
    <o-dd:software>
      <o-ex:context>
        <o-dd:version> 7.0 </o-dd:version>
        <o-dd:uid> oma:real:helix </o-dd:uid>
      </o-ex:context>
    </o-dd:software>
  <o-ex:constraint>
```

</o-dd:move>

The issue of exporting requires greater analysis as to the many options that content maybe exported to. Then the semantics can be further refined by reusing ODRL terms and potentially defining new semantics.

V. INHERITANCE MODEL

The inheritance model adopted by OMA DRM is aimed at supporting the "subscription" business model. It uses a Parent Rights Object (RO) as the key to any subsequent Child ROs that are delivered to the device. In effect, you need the Parent RO for the Child RO to "inherit" from - and this requirement - is mapped into being a member of a subscription service.

This is an interesting take on the original need for an inheritance model in ODRL. This was based on the more traditional need to generically inherit rights from other rights statements.

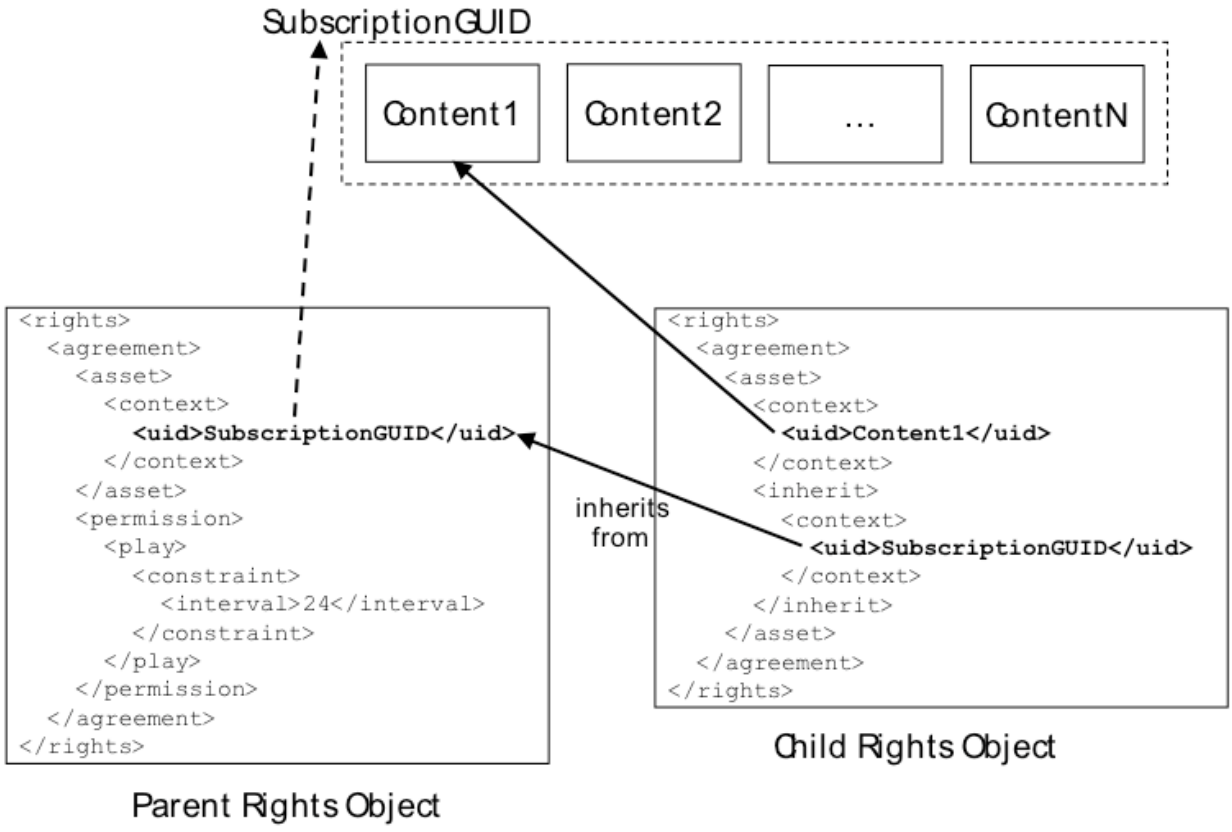
The OMA DRM view on the inheritance model is to overload the UID element. In this case, if the UID refers to some "virtual" content, and the Child ROs inherit from the same UID, then we have a subscription model. See the Figure below taken from the OMA DRM REL Specification [4].

The SubscriptionGUID has special meaning and a DRM Agent must be able to detect this. This is helped by the fact that a Parent RO will not have a KeyInfo or Digest element, as there is no real content. However, the subscriptionGUID does "point" to a real thing - so one could argue that all the Content (from 1 to N) inside this thing - are all available from just the Parent RO only.

A number of questions arise from this subscription inheritance model:

- Where should the actual permissions be located?
- What about the current state of the inherited permissions?
- Is it needed at all?

There are three models as to where the actual permissions may be located. In the Parent RO, Child RO, or both. In most cases, the Child RO should contain the actual permissions, as this is "closer" to the content, and Parent RO is really being used as a further "check" that the client has previously subscribed to this service. This a little bit of "over kill" as a Child RO, as a normal RO, that is sent to a client, would require some back-end service knowing who has already subscribed and will be pushing out the subsequent content -



all cryptographically bound to the end device. Nevertheless, the use of inheritance does make the model seem more “realistic”.

Technically, the permissions may appear in any of the three model options above. Experience over time will tell if any of the models have greater benefits over the others.

Another issue deals with the current state of the rights expression. When you inherit, do you not only inherit the permissions, but their current state? Is this desirable as well? The ODRL specification is silent on this issue, but the original intent was that you only inherit the permissions, not their current state.

The example in Table 1 shows the Parent RO with an Interval constraint (we assume the value should be “P24H”). This may mean that the total subscription covers a 24 hour period - or that each part of the subscription covers a 24 hour period. What would happen if the “state” of the Parent RO was expired (ie a 24 hour period had passed since it was first used) and then a Child RO is received with no new permissions? It could be interpreted to mean that the whole subscription is finished (based on the Parent RO) or that you now have another 24 hours to play the new content.

The various options here will need to be more fully discussed. There probably are cases where the example described may be the desired outcome. (For example, the OMA DRM specification makes a clear decision that state is not copied when exporting ROs. The same maybe needed for inheritance.)

The last big issue is if inheritance is needed at all to support subscription. You could certainly use inheritance for the original idea of having common rights that other expressions can use to inherit from. But there is also nothing stopping a service from providing a subscription to content and simply sending ROs when appropriate to the client.

VI. UIDs AND VERSIONS

The specification uses both the UID and Version context elements to define which systems to limit any exports to. The Open Mobile Naming Authority (OMNA) - part of OMA - will publish formal identifiers for the various systems. However, there maybe some need to standarise on the version numbering as well. For example, even the simple difference between “10” and “10.0” may make a difference to the parsing of the version number. Even worse may be non-numerical version identifiers.

VII. OVERRIDING SEMANTICS

The specification indicates: “If the <export> permission is granted to more than one target system, then these are enumerated by using multiple <uid> elements. In this case, the <count> constraint applies to the combined export transactions of all target systems.”

This has overridden the normal semantics of ODRL. In the normal case, a count constraint would be “and-ed” with all the other constraints. So, a count of “1” for two “systems” would allow both to occur.

For the OMA DRM view to be expressed (in this particular case), you can use the Container construct with the “or” boolean between two system constraints.

We assume that by doing this (supporting the container model) would increase the complexity of processing the ROs.

VIII. GRACE PERIOD

The Interval and Accumulated constraints both must “stop the execution of the permission as soon as possible after the value of the element has elapsed” and that this “should happen immediately”. It is not clear why “as soon as possible” was included nor why the “should” is not a “must”.

There now seems to be some possible "delay" to the Accumulated and Interval constraints. And in some cases, this could be a “user friendly” issue and by offering some “grace period” would improve the DRM experience.

This could then be generalised with the "timer" attribute in <timed-count>. That is, have one attribute "oma:delta" that indicates the number of

seconds you can wait before the permission must stop or be recorded.

IX. PRIVACY ISSUES

The OMA DRM specification allows for content to be shared - forwarded from user to user via super-distribution. This transaction can be tracked by the implementation. The OMA DRM specification does not include tracking as part of the REL - even though that facility is available in the complete ODRL REL. In such cases "tracking" can become one of the ODRL Requirements, hence explicitly making this feature something that the consumer has to agree to before acquiring the content.

At the same time, this also make it clear to the consumer what will happen when then do acquire this content. Since OMA DRM leaves tracking to be an "implementation issue", it does not guarantee that consumers will be aware of this requirement. Worse, it creates an nebulous situation in which a consumer's actions can be reported without their clear knowledge. In some cases, this can lead undesirable outcomes for the end consumer.

OMA DRM should have included the "tracked" requirement in their REL profile. This would make it always clear to the consumer what they can expect, and ensure that their privacy is not compromised.

X. OTHER ELEMENTS

The new profile did not consider including the rights holder and payment information. The rights holders would have been useful to assert the true owners of the content and may then allow end users to be aware that there is such important information available.

Perhaps the most disappointing is the non-inclusion of payment information. To meet the long term goals of interoperable content services, there needs to be support for information on how payments are handled. This would enable content owners to provide packaged content+rights to many different service providers, and not have to deal with each individually on the terms and conditions for payments.

XI. CONCLUSION

This paper has reviewed some of the decisions made by the OMA DRM Working Group in developing the ODRL profile for version 2.0. It has provided some feedback towards different options that may have been available, as well as discussed some of the advantages and disadvantages of these decisions and raised some of the semantic issues.

Overall, I think this is an excellent use-case for all parties (OMA and the ODRL Initiative) and will help in future work and more specifically, future ODRL profiles for OMA DRM.

ACKNOWLEDGMENT

National ICT Australia (NICTA) is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs.

REFERENCES

[1] Iannella, R. The Open Digital Rights Language: XML for Digital Rights Management . Information Security Technical Report, Volume 9, Issue 3, July-September 2004, Pages 47-55.

[2] OMA (2003). Open Mobile Alliance DRM Specifications, Version 1.0 Approved Enabler, November 2003 <http://www.openmobilealliance.org/release_program/drm_v10.html >

[3] OMA (2004). Open Mobile Alliance DRM Specifications, Version 2.0 Candidate Enabler, July 2004 <http://www.openmobilealliance.org/release_program/drm_v20.html >

[4] OMA (2004). Open Mobile Alliance DRM Rights Expressions Language Version 2.0, 10 Dec 2004 <http://www.openmobilealliance.org/release_program/docs/DRM/V2_0-20041207-C/OMA-DRM-REL-V2_0-20041210-C.pdf>