27-10-2006

Sharing Digital Rights with Domain Licensing

N. P. Sheppard
University of Wollongong, nps@uow.edu.au

R. Safavi-Naini
University of Wollongong, rei@uow.edu.au

Sharing Digital Rights with Domain Licensing

Abstract

Sharing of multimedia content is a common practice that, combined with appropriate business models, need not be detrimental to the interests of content providers. Existing digital rights management systems, however, support only relatively limited sharing of content between multimedia terminals, resulting in inconvenience and frustration for end-users of rights-managed content. In this paper, we propose to combine the notion of an "authorised domain" with an "environment role" to permit end-users to share access to multimedia content within the constraints expressed in a domain licence. We describe how a variety of different business models can be supported using domain licences, and propose a preliminary domain expression language in which licences can be written. Finally, we demonstrate the practicality of our model by outlining how it could be implemented using the Open Mobile Alliance's specification for authorised domains together with a ubiquitous computing network. Our proposal provides greater expressive power than the base OMA DRM framework without requiring users to upgrade their devices.

Keywords

digital rights management, ubiquitous computing, open mobile alliance, rights expression languages

Disciplines

Physical Sciences and Mathematics

Publication Details

This conference paper was originally published as Sheppard, NP and Safavi-Naini, R, Sharing digital rights with domain licensing, Proceedings of the ACM Workshop on Multimedia Content Protection and Security, Santa Barbara, USA, 27 October 2006.

ACM Workshop on Multimedia Content Protection and Security 2006

Sharing Digital Rights with Domain Licensing*

Nicholas Paul Sheppard Reihaneh Safavi-Naini

November 10, 2006

Abstract

Sharing of multimedia content is a common practice that, combined with appropriate business models, need not be detrimental to the interests of content providers. Existing digital rights management systems, however, support only relatively limited sharing of content between multimedia terminals, resulting in inconvenience and frustration for end-users of rights-managed content. In this paper, we propose to combine the notion of an "authorised domain" with an "environment role" to permit end-users to share access to multimedia content within the constraints expressed in a *domain licence*. We describe how a variety of different business models can be supported using domain licences, and propose a preliminary *domain expression language* in which licences can be written. Finally, we demonstrate the practicality of our model by outlining how it could be implemented using the Open Mobile Alliance's specification for authorised domains together with a ubiquitous computing network. Our proposal provides greater expressive power than the base OMA DRM framework without requiring users to upgrade their devices.

1 Introduction

The increasing availability of network technologies has made electronic distribution an attractive mode of distribution for multimedia content due to the convenience and low cost of copying and distributing digital multimedia. However, this convenience applies equally to legitimate and illegitimate distribution channels, and fears of widespread copyright infringement and other missues of multimedia content are often blamed for the slow uptake of electronic distribution by content owners.

Digital rights management ("DRM") allows content owners to control and monitor the distribution of multimedia content through electronic channels using a machine-enforceable *licence*. Content owners' fears of widespread copyright violation via electronic distribution have seen digital rights management become a fast-growing field of research and development in recent years, and a number of systems are now commercially available.

Digital rights management systems, however, do not always provide end-users of multimedia content with the experience they have come to expect from older models of multimedia distribution. In particular, while many modern digital rights management systems allow some restricted sharing of content amongst a small group of multimedia terminals, these systems lack the simplicity and flexibility formerly offered by physical media that could be easily transported between terminals, swapped amongst friends, stored in libraries, and so on.

In this paper, we propose a model for sharing access to multimedia content based on context-aware access control and the *authorised domain* concept proposed by the Open Mobile Alliance [20], Digital Video Broadcasting Project [10] and others. Our model allows access to multimedia content to be shared amongst a pool of users and devices, within limits defined by the content provider and enforced

by a context-aware access control system.

We will give an overview of existing proposals for authorised domains and context-aware access control in Section 2, then describe our rights-sharing model in Section 3. Our model augments the authorised domain concept with a *domain licence* that controls the form of a domain. A domain licence sets out the contexts in which a terminal may be a member of a domain and therefore have access to the pool of content belonging to that domain. Domains can be set up, for example, to encompass a household, a sporting event, or the set of devices belonging to a user.

We will describe how our model maps to a variety of sharing scenarios in Section 4, and derive the context information required to implement these scenarios. We will then propose a preliminary *domain expression language* in Section 5 that can be used for writing domain licences supporting our example scenarios and others.

Finally, we demonstrate the practicality of our model by outlining how it could be implemented using the Open Mobile Alliance's digital rights management specification in Section 6. Our proposal greatly increases the expressive power of OMA's digital rights management framework without requiring an upgrade to end-user devices.

2 Previous Work

2.1 Authorised Domains

Many existing digital rights management systems recognise that users typically have more than one terminal with which they would like to access their content, and support some primitive method of sharing content amongst a small group of terminals. The Digital Video Broadcasting ("DVB") Project [10], Secure Video Processor ("SVP") Alliance [25], Open Mobile Alliance ("OMA") [20] and Marlin Developer Community [15], for example, all intend for their content protection standards to incorporate "authorised domains", that is, collections of terminals that have access to a particular pool of content.

Schemes for authorised domains proposed by Marlin, OMA, TIRAMISU [16], Popescu, et al. [22] and Sovio, et al. [26] have domains being defined by some device or Internet service that we will refer to as being the *domain controller*. A terminal may join a domain by contacting the domain controller that created the domain and, once it has done so, it can access content that belongs to the domain.

None of these schemes, however, define the method by which the domain controller decides whether or not to allow a terminal to join a domain. In this paper, we propose a model in which a domain controller can be provided with a description of a domain that it will use to determine which terminals should be members of that domain. We will outline how this model could be implemented using OMA domains in Section 6.

Authorised domain systems proposed by IBM [21], Thomson [1] and the SVP Alliance [25] allow devices to be joined to a domain up until a limit on the size of the domain is reached. Koster, et al. [13] and the Marlin specification hint at the notion of a domain licence (called a "domain policy") used in this paper, but support only simple policies based on the domain's size and the proximity of the proposed member to the domain controller. These simple policies may be sufficient in the household sharing applications for which these systems are designed. However, this model does not seem sufficient for all kinds of domains and in this paper we propose a much more complete method of defining domains.

Reddy, et al. [23] describe a system in which groups of users are defined by lists of individual users maintained by a *licence device* (that is, "rights issuer" in the terminology used by this paper). In order to obtain access to a particular object, a user must contact the licence device in order to check that he or she is on the list of people permitted to access this object. This procedure is used by Microsoft's Rights Management Services [18]. In the model used in this paper, a single licence may be shared by a collection of devices without each one needing to contact the rights issuer individually.

2.2 Context-Aware Access Control

This paper proposes to use context information about a multimedia terminal in order to determine whether or not it should be permitted to join a domain. Many "context-aware" access control systems have been developed for ubiquitous computing applications in which access to some resource depends on context information. Context-aware access control models can be divided into two kinds:

- models in which context information is checked as an additional component of an access control rule [9, 5, 8]; and
- models in which context information is used to permit activation of an "environment role" to which permissions are assigned [6, 24, 2, 11].

The approach used by digital rights management policy languages such as the Open Digital Rights Language ("ODRL") [19] and MPEG Rights Expression Language ("MPEG REL") [12] can be thought of as an example of the former kind. In these languages, access to a resource can be made conditional upon a variety of context information including time and location.

In this paper, however, we propose to make membership of an authorised domain akin to membership of an environment role. This approach allows context information to be processed by a relatively powerful central domain controller while individual terminals need only know how to join and leave a domain and perform a few relative basic cryptographic functions. In particular, sophisticated context-aware constraints can be supported using legacy terminals that do not otherwise support such constraints.

Unlike environment roles, however, membership of our authorised domains is not automatic upon satisfying some context. In general, membership must be further authorised by a *domain controller*. This allows content users to control which terminals enter their domains within the constraints imposed by the content provider. For example, if a content provider has limited the size of an authorised domain to ten terminals, the domain controller may choose which ten terminals these are and need not simply take the first ten terminals that ask to join.

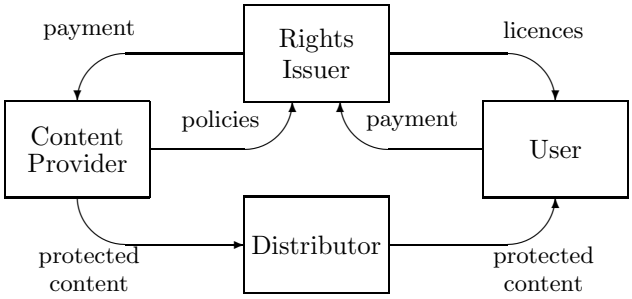

Figure 1: A typical digital rights management system

3 Sharing Digital Rights with Domain Licensing

3.1 Traditional Digital Rights Management

Figure 1 shows the architecture of a typical digital rights management system. Content is created by a *content provider*, and transmitted in a protected (for example, encrypted) form to a *content user* via some distribution channel. In order to access the protected data, the user must obtain a *licence* from the *rights issuer*.

Licences are written in a machine-readable *rights expression language* that sets out the terms of use of the data and the information required to access the protected content. In the model proposed by Guth, et al. [7], a licence is composed of a collection of *licence objects* of one of five types:

Resource. A unique object associated with some identifiers.

Subject. A party to the licence, being either a *rights-holder* who owns the rights over a resource, or a *beneficiary* to whom rights to access a resource can be granted. Each subject is associated with a unique identifier.

Role. A group to which subjects may belong and to which permissions may be assigned, as in role-based access control.

```
<rights>
  <agreement>
    <party>
      <context>
        <uid>
          x500:c=AU;o=XYZ;cn=Alice Jones
        </uid>
      </context>
    </party>
    <asset>
      <context>
        <uid>
          http://www.videos.com.au/trailers
        </uid>
      </context>
    </asset>
    <permission>
      <play>
        <constraint>
          <count>10</count>
        </constraint>
      </play>
    </permission>
  </agreement>
</rights>
```

Figure 2: A licence allowing Alice Jones to play ten video trailers from `http://www.videos.com.au/trailers`

Permission. The right to perform a particular operation on a resource.

Constraint. A conditional expression that must be satisfied in order for some permission to be exercised.

A licence is an aggregation of permissions awarded by some rights-holder to some beneficiary. Licences can only be issued on the authority of the rights-holder and contain the information required to decrypt the protected form of the resource specified in the licence. A (simplified) example of a licence written in the Open Digital Rights Language [19] is shown in Figure 2.

The fundamental security requirement of a digital rights management system is that terminals be guaranteed by their manufacturers to behave in accordance with licence documents. For the purposes of this paper, a *terminal* is an abstract single-user multimedia player, editor, etc. that may be implemented as a hardware device, a software application or combination of the two. Note that, on a multi-user device such as a mainframe computer, it may be possible for two or more software terminals to be concurrently executing on the same machine and it is important that two such terminals be considered as independent entities.

When a user (beneficiary) wishes to perform some particular operation on a particular resource, the terminal checks that the user possesses a licence granting that permission, and that any constraints associated with the permission are satisfied. If the permission does not exist, or the constraints are not satisfied, the terminal will refuse to carry out the operation. Otherwise, the terminal will retrieve the content key and carry out the desired operation.

Licences will only be accepted by the terminal if they are signed by a valid rights issuer recognised by the terminal. Rights issuers only generate licences according to some policy set by the relevent content provider; for example, users must pay a certain amount of money in order to obtain a licence for a particular item of content.

3.2 Domains

We call the fundamental sharing mechanism in our model a *domain*, as in other work. A domain is an arbitrary collection of terminals. A domain may be created by anyone, and all authority to manage the domain and issue resource licences for it is derived from the creator of the domain.

The beneficiary of a resource licence is a domain. Any terminal that is a member of the beneficiary domain of a resource licence may exercise the permissions granted by the licence, subject to any constraints specified by the licence.

Every terminal is a member of a *unit domain*, which is the domain containing that terminal and no other. The *universal domain* is the domain that contains all terminals that are compliant with the digital rights management regime. A terminal may

be the member of an arbitrary number of domains simultaneously, so long as it satisfies the criteria for belonging to all of those domains.

3.3 Domain Licences

A *domain licence* is a licence that sets out the terms under which a terminal may be a member of a domain. As for resource licences, domain licences may only be issued on the authority of the creator of the target domain. Following the model of Guth, et al. for resource licences, a domain licence is composed of a collection of licence objects:

Target Domain. The domain whose membership is controlled by the licence, analogous to the resource in a resource licence.

Beneficiary Domain. The domain whose members are permitted to make terminals a member of the target domain, analogous to the beneficiary in a resource licence.

Membership Criterion. A conditional expression that must be satisfied in order for a terminal to be an active member of the target domain, loosely analogous to the constraint in a resource licence. Note that satisfaction of the membership criterion may not automatically confer domain membership on a terminal; the terminal must still obtain approval from a member of the beneficiary domain (which, in general, may be refused).

Domain licences are written in a *domain expression language* analogous to the rights expression language used to express resource licences. As for resource licences, domain licences must only by accepted by a domain controller if they are signed by a domain creator recognised by the domain controller. We will consider domain expression languages in more detail in Section 5.

3.4 Sharing Constraints

Constraints in both resource and domain licences may be *stateful* or *stateless*. Stateless constraints are constraints whose applicability can be checked without any memory of previous operations on the associated resource; for example, a constraint specifying a time interval during which the resource can be accessed is stateless. Stateless constraints in domains behave in the same way as they do in traditional digital rights management.

Stateful constraints are constraints whose applicability can only be checked with reference to state information that is updated every time an operation is performed on the resource. For example, a constraint that specifies the number of times that a resource can be used is stateful. We identify two ways in which stateful constraints can be applied in domains:

Shared. Each domain has a global state that is affected by the actions of all of the members of the domain. For example, a domain has a pool of credit that is consumed whenever any member of the domain uses a resource.

Unshared. Each member of the domain maintains state information independently of all other members of the domain. Each member's state information is affected only by the actions of that member. For example, each member of the domain may use a resource once.

Both types of stateful constraint seem useful. Every instance of a stateful constraint in a shareable digital rights management system will therefore be associated with a "shared" flag indicating whether or not this constraint is to be applied relative to the state information held in common by the beneficiary domain, or by the particular domain member that exercises the associated permission. Constraints in the Open Digital Rights Language, for example, are shared by default but the language also supports an attribute called `forEachMember` that indicates that the associated constraint is unshared.

The introduction of domains allows the introduction of new kinds of stateful resource constraints in which the "state" refers to some state of the domain itself. For example, there may be a limit on the number of domain members permitted to access a particular resource at any one time. We will refer to these kinds of contraints as *domain-state* constraints.

3.5 Retention

If a domain criterion specified in a domain licence ceases to be satisfied, the terminal must leave the target domain. When a terminal leaves a domain, the terminal becomes a *lapsed member* of that domain. Lapsed members may not exercise any permissions granted to the domain.

Every domain may be associated with some state information, including stateful resource constraints and stateful domain membership criteria. Lapsed members must continue to store any state information associated with the domain, and re-activate this information should they re-join the domain at a later date. This prevents dishonest users from resetting state information by artificially causing their terminals to leave and re-join a domain.

A real terminal will eventually run out of space if it joins many domains that require it to store state information indefinitely. In order to allow terminals to reclaim storage space used by ephemeral domain memberships, a domain licence may contain a *retention criterion*. The retention criterion is a conditional expression that, if satisfied, requires a lapsed member of the associated domain to retain the state information related to that domain. Conversely, if the retention criterion ceases to be satisfied, a lapsed member of the domain may delete the state information. We will refer to this operation as *deletion* of a domain.

4 Sharing Scenarios

We will now consider some example scenarios for sharing digital resources. It is not practical to describe every conceivable scenario in which resources might be shared here, but we have attempted to design a variety of representative and realistic scenarios. We will show how these scenarios can be mapped to our model of sharing and identify the context information required to implement these models of sharing. This context information will need to be supported by the domain expression language we will outline in Section 5, and the implementation we will outline in Section 6.

4.1 Social Sharing

CDs, DVDs and the like are commonly shared amongst social networks as a means of socialisation and exploring new music and film [3]. We are not aware of any model of a social network that would be meaningful to a computer. However, a simple but useful model might be to associate every human user with an *acquaintance domain*, membership of which is controlled by some device or set of devices owned by that user. Another device may be a member of a user's acquaintance domain if the owner of the device is acquainted in some sense with the owner of the acquaintance domain. "Acquaintance" can be defined in a variety of ways, including

- allowing each user to nominate a fixed number of acquainted devices;
- fixing the maximum number of acquaintance domains that any one device can be a member of;
- requiring an acquainted device to be in frequent close physical proximity to the domain controller; or
- by referring to registered relationships such as employment or marriage.

Context Required. The number of terminals in a domain; the number of domains of a particular type of which a terminal is a member; physical location; external relationships.

4.2 Site Licensing

Many software vendors provide site licensing packages that allow businesses and other institutions to make a software package available to its employees. In our model, the software vendor would issue a domain licence for a domain whose membership depends on whether or not the proposed member belongs to the organisation. This domain licence might be issued to a software server that acts as the domain controller for the domain. Machines that belong to the organisation might be

- listed explicitly in the domain licence;

- identified by their membership of the organisation's local area network;
- identified by their physical location within the organisation's offices;
- identified by some combination of the above three; or
- the domain licence might simply limit the number of terminals on which the software can be installed.

Context Required. The number of terminals in a domain; physical location; network location; machine identity.

4.3 Household Sharing

Consider a family household. Systems such as those described in Section 2 are designed to implement sharing amongst all of the members of a household by creating a domain containing all of the terminals in the household. Resource licences are then issued to this domain.

Using domain licences, it is possible to control membership of the household domain in a more sophisticated way than by simply placing an upper limit on the number of terminals that can be in a household. A household can be defined by the building it occupies; by a role containing all of the members of the family; or by the local area network within the household. A device fixed within the home such as a set-top box or media centre PC might act as a domain controller.

Sub-role domains (corresponding to sub-roles) can be used within the household to restrict access to material (such as personal photo collections) that is private to one member of the household, or to material (such as violent or pornographic material) that may not be appropriate for all members of the household.

Context Required. Membership of a super-domain; physical location; network location; user identity and role.

4.4 Club Sharing

Consider a film club. Club members pay a subcription fee to the club that allows them to download films from a group of libraries with whom the club has an agreement. Membership of the club can be seen as equivalent to membership of a domain whose membership criterion contains a payment predicate.

Furthermore, suppose the club organises monthly public screenings in which films are projected onto a large screen in an open field. Patrons – who may or may not be club members – may watch the film from their cars in a carpark at the back of the field, or sit on the grass nearer to the screen. To avoid upsetting the surrounding residences, however, the "cinema" is not equipped with loudspeakers.

The soundtrack for the film is available via a wireless network created by the film projector. In order to listen to the soundtrack, patrons must connect to the network using their own terminals (such as in-car and handheld players). The soundtrack is made available to a domain, of which patrons' terminals can become members by contacting a domain controller hosted alongside the film projector.

In order to recover the costs of hiring the field and showing the film, the club sells access to the soundtrack domain. Patrons who are not club members must pay a surcharge that the club keeps and uses to buy new equipment, etc. The soundtrack domain can be implemented using two domain licences, one that requires the target terminals to already be a member of the club's domain, and another that anyone can join. The former licence requires the payment due from club members, while the latter requires the payment due from non-members.

Context Required. Membership of a super-domain; physical location; payment.

5 A Domain Expression Language

The power and flexibility of our model depends to a large extent on the expressiveness of the domain expression language in use – the universe of domains that can be supported by our model is governed by

the expressiveness of the domain membership criterion.

Covington, et al. [6] use a Prolog-based language to describe the activation conditions for environment roles, but do not specify the fundamental predicates from which environment roles can be built in their system. That is, they define the syntax of the language but do not define a specific vocabulary of criteria that can be used to define environment roles.

Masone proposes a "role definition language" based on mathematical set notation, SQL and C [17]. Masone's language is bound to a ubiquitous computing system known as "Solar" [4] in which sensors transmit a series of events to other nodes in the network. Masone's language allows a security designer to specify a collection of sets (some of which represent environment roles) that are updated whenever a security-sensitive event occurs. Writing role definitions for a particular network appears to require detailed knowledge of the network that is unlikely to be available to the author of a domain licence.

In digital rights management applications, it needs to be possible for a content provider to describe a domain in broad but unambiguous terms that can be interpreted for any network in which the content might be used. Individual networks may use approaches similar to those of Covington, et al. or Masone to check the predicates of a domain expression language such as "terminal X is located in room Y" or "user A is logged on to terminal B"; we will consider this further when we outline a possible implementation in Section 6.

In this section, we will present a preliminary domain expression language based on our observations about domain licensing in the foregoing sections. The syntax and vocabulary of our language is based on that of the Open Digital Rights Language [19], which is used as the rights expression language in OMA's digital rights management system. Other syntax and vocabulary could be used though we expect the differences would be largely cosmetic.

ODRL, and our domain expression language, are based on XML. We will call the root element of a domain licence `rights`, as in ODRL. Each domain licence may contain one or more `agreement` elements that describe an agreement between the content provider and the beneficiary domain for operating a domain. Each agreement contains an element `party` identifying the beneficiary domain; an element `target` identifying the target domain; an element `membership` identifying the membership criterion and, optionally, an element `retention` identifying the retention criterion. For simplicity, we do not consider an equivalent of the ODRL `offer` or `revoke` elements in this paper.

5.1 Beneficiary and Target Domains

Domains can be identified using a `uid` ("unique identifier") element within the `party` and `target` elements using the `context` element. Any method of generating a unique identifier is suitable.

5.2 Membership and Retention Criteria

Both the membership and retention criteria may contain an arbitrary number of elements representing one predicate of the criterion. If every predicate contained by the criterion is true, then the criterion is satisfied. Disjunctive relationships between predicates can be expressed by using multiple `agreement` elements with the same beneficiary and target domains, but different membership and retention criteria. Predicates can be negated by enclosing them within the `not` element.

Most predicates are likely to be useful for expressing both membership and retention criteria, and we will allow the use of the same set of elements within both criteria. For clarity, we will give an interpretation of each predicate in both contexts.

We will refer to a terminal that is being considered for membership of a domain as the *target terminal*. Membership and retention criteria will be evaluated against the context of the target terminal.

5.2.1 Simple Criteria.

Many of the context elements identified in Section 4 represent simple concepts such as size and location. While implementing a system for securely checking these criteria may be non-trivial, a variety of systems

for doing so have been described in the literature and we will not consider the detail of these systems in this paper. Many of these concepts are represented by elements in ODRL that we will adopt in our domain expression language. We also adopt a few elements of ODRL that, though they do not appear in any of our scenarios, seem useful to express widely-used concepts such as expiry dates.

Accumulated. Terminals may only accumulate a given amount of time as a member of the target domain; or, must retain the domain until a given amount of time has been accumulated as a lapsed member.

Count. The number of terminals in the domain must not exceed a given value. This seems useful only as a membership criterion.

DateTime. Terminals may only be a member of the domain during a given time period; or, may delete the domain once a given time has passed.

Network. Members must be located within a given computer network; or, must retain the domain until they have left the given network.

Spatial. Members must be physically located within a given spatial region; or, must retain the domain until leaving the given region.

Payment. A fee must be paid in order to add members or to delete a domain.

Tracked. Every join operation or domain deletion will be logged.

5.2.2 Domain Criteria.

The `domain` predicate is true if the target terminal is a member of the domain specified by the `context` element within the `domain` element. This predicate can be used to create hierarchical domains in which the target domain is a sub-domain of the domain identified by the `domain` predicate, and also to create mutual exclusion relationships between domains by negating the `domain` constraint.

Domains may belong to a *domain class* such as personal domains and acquaintance domains. It is often useful to create mutual exclusion relationships between domains of one class such as "a terminal may be a member of at most one personal domain at a time" or "a terminal may be a member of at most ten acquaintance domains at a time". This can be expressed conveniently in a predicate `class-multiplicity` that contains the maximum number of domains of that class that a terminal may be a member of at any one time.

Note that these semantics make it possible for one domain licence to cap multiplicity of a domain class to one number, and another licence to cap multiplicity of the same domain class to another number. Hence, it may be possible for a terminal to belong to a larger number of domains of the same class if one set of licences is exercised as compared to another set for the same domain class. This may confuse users and it would be possible to prohibit such a state of affairs (for example) by requiring that each domain be controlled by only one licence, but we leave it up to domain designers to develop good policies for their applications.

5.2.3 Role Criteria.

Human users may occupy *roles* in the sense of a role-based access control system. We assume that terminals have some method of authenticating their human users that is beyond the scope of this paper, and similarly of checking whether or not the user is the member of a particular role.

A *role domain* is a domain whose membership depends on the current human user of the terminal. That is, a terminal may only be a member of a role domain if its current user is a member of the corresponding role. As in ODRL, we use the `group` predicate to express this in a membership criterion.

The *universal role (domain)* is the role (domain) containing all humans, and every user has his or her own *unit role (domain)* that contains that user and no other. We will also sometimes refer to the unit role domain of a particular user as being his or her *personal domain*; personal domains are very similar to the "personal entertainment domain" proposed by Koster, et al. [13] and contain all of the devices used by a particular human user. As in ODRL, the `ind-`

```
<rights>
  <agreement>
    <party>
      <context>
        <uid>alice:controller</uid>
      </context>
    </party>
    <target>
      <context>
        <uid>alice:acquaintance</uid>
      </context>
    </target>
    <membership>
      <class-multiplicity>10</class-multiplicity>
    </membership>
    <retention>
      <accumulated>P30DT</accumulated>
    </retention>
  </agreement>
</rights>
```

Figure 3: A domain licence for an acquaintance domain.

`ividual` predicate can be used to refer to a single human user, that is, the owner of a unit role.

5.3 Examples

Figures 3 and 4 show examples of domain licences written in our domain expression language. We have omitted the signatures on these licences for brevity; the licence of Figure 3 would probably be signed by some authority recognised by all of the content providers who use the system, while the licence of Figure 4 might be signed by the film's distributor.

Figure 3 shows a licence that allows Alice's domain controller (`alice:controller`) to make another terminal a member of Alice's acquaintance domain (`alice:acquaintance`). The incoming device must not be a member of more than nine acquaintance domains already, and may delete the domain if it spends more than thirty days outside the domain.

Figure 4 shows a licence implementing one case of the club sharing scenario described in Section 4.4. Here, club members' terminals are assumed to be members of a domain called `myfilms`. The domain controller `myfilms:controller` may join a member of the `myfilms` domain to a sub-domain called `myfilms:2006:2`, which is the domain used for the second film session of the year 2006. In order to be permitted to join the sub-domain, terminals must be located in Oak Tree Park and must pay three dollars. Since the domain only exists for the showing of the film, terminals may delete it from their memories once the film has concluded.

6 Implementation Using OMA Domains

We will now outline how the model described in Section 3 can be implemented using OMA domains. OMA DRM provides a specification for devices (principally, mobile phones) that can be members of domains and make use of content granted to those domains, but does not specify how the domains themselves are governed.

OMA DRM supports two kinds of *rights objects* (resource licences), which we will refer to as *device rights objects* and *domain rights objects*. The rights given in a device rights object may only be exercised by the device identified in that rights object, whereas the rights given in a domain rights object may be exercised by any member of the domain specified in the rights object.

Rights objects of either kind can be issued by *rights issuers*. Rights issuers are also responsible for determining whether or not a device is permitted to join a domain. In general, rights issuers may create and manage domains according to their own whims.

For the present purposes, however, we consider a special kind of rights issuer, called a *domain controller*, that manages domains according to domain licences issued to it by some *master rights issuer* that creates those domains. Domain controllers must meet tamper-resistance requirements comparable to those met by end-user devices that guarantee that they will behave in accordance with the licences they have been given. Every domain must have exactly one domain controller, but a single domain controller may man-

```
<rights>
  <agreement>
    <party>
      <context>
        <uid>myfilms:controller</uid>
      </context>
    </party>
    <target>
      <context>
        <uid>myfilms:2006:2</uid>
      </context>
    </target>
    <membership>
      <domain>
        <context>
          <uid>myfilms</uid>
        </context>
      </domain>
      <spatial>Oak Tree Park</spatial>
      <payment>
        <amount currency="AUD">3.00</amount>
      </payment>
    </membership>
    <retention>
      <datetime>
        <end>2006-01-14T23:59:59</end>
      </datetime>
    </retention>
  </agreement>
</rights>
```

Figure 4: A domain licence for club members.

age an arbitrary number of domains.

Domain controllers and master rights issuers both implement the role of a rights issuer in the protocols described by OMA. Master rights issuers, however, are additionally permitted to issue domain licences to domain controllers using a trivial variation of OMA's Rights Object Acquisition Protocol. Domain controllers are permitted to execute OMA's protocols with end-user devices only in accordance with licences that have been issued by the master rights issuer.

Domain controllers are required to be able to access relevent context information about the devices that may join their domains, and be able to verify that this information is correct. Numerous methods for obtaining context information have been developed by the ubiquitous computing community, though not all of them were designed with security in mind. In this paper, we adopt the model used by Masone [17] and assume that the domain controller receives a trusted series of events describing context changes.

For every domain that it controls, the domain controller maintains a set of devices that satisfy the membership criterion of that domain. The terminals in this set may or may not be actual domain members. Whenever a context event is received, the domain controller updates the sets according to some event handler.

If a set gains a new member, the domain controller may choose to invite the incoming device to the domain by initiating OMA's Join Domain protocol. The human user of the device may veto the invitation if he or she does not wish the device to become a member.

If a set loses a member, the domain controller can ask the outgoing device to leave the domain by initiating OMA's Leave Domain protocol. It may not be possible to force the device to leave the domain immediately, however, since the OMA specification requires that the user of a device be able to veto a request to leave a domain, or the device may have gone off-line. Users can, however, be encouraged to leave domains by being refused any further service until they come back on-line and leave the domain voluntarily.

The human user of a device can also request to join or leave a domain by initiating the Join or Leave

Domain protocol from the device's side. If the user requests his or her device to join a domain, the domain controller checks that the device is a member of the corresponding set. If it is, the domain controller may permit the device to join the domain. Otherwise, the request is rejected. Note that the domain controller may refuse to admit a device to a domain for its own reasons, even if the device satisfies the membership criterion for the domain.

We will now consider how different elements of the model proposed in Section 3 can be implemented using this architecture.

6.1 Simple Criteria

It is straightforward to implement the simple criteria suggested in Section 5.2.1 so long as the ubiquitous computing network is able to report the relevent context information to the domain controller. In some cases – such as obtaining the current time or obtaining the current number of devices in a domain – this is trivial.

Verification of criteria such as `spatial`, `network` and `payment` is somewhat more involved. Many systems have been developed for locating devices and accepting electronic payment, and many of these could be incorporated into our system. We will not consider these systems in detail here.

6.2 Domain Criteria

OMA does not define any way in which a rights issuer (domain controller) can query a device about which domains it already belongs to. However, since devices can only join and leave a domain by interacting with that domain's controller, it is straightforward for each domain controller to maintain a list of all of the members of all the domains that it manages.

If a domain licence contains a `domain` criterion, the domain controller of the target domain must query the domain controller of the specified domain as to whether or not the target device is a member of that domain. This may require the domain controller of the target domain to consult some register of domain controllers in order to locate the controller of the specified domain. There are several straightforward ways in which this might be implemented and we will not consider the details of such a register here.

6.3 Role Domains

OMA does not provide any support for authenticating the user of a device, so it is not immediately obvious how role domains might be implemented. One approach, however, is to implement personal domains by issuing a domain licence with a `count` criterion of one. In this way, only one device (presumably the one that is currently in use by the user) may be active in the domain at any one time.

In order to prevent attackers from enrolling their own devices in the user's domain, the master rights issuer must choose domain identifiers that are difficult to guess, and users must keep their domain identifiers secret. Identifiers of this sort could be produced, for example, by having the master rights issuer choose unique plaintext identifiers and encrypting these using a secret key in order to produce ciphertext identifiers assigned to users.

Given that personal domains have been implemented in this way, role domains can be implemented by issuing domain licences whose membership criterion includes a `domain` criterion listing all of the personal domain identifiers of the members of that role.

Using this technique, it is not necessary for the domain expression language to explicitly support the `group` and `individual` criteria we suggested in Section 5.2.3.

6.4 Retention

OMA does not require that domain rights objects and content be deleted when leaving a domain. These rights objects and content will become immediately re-available when the device re-joins the domain at a later time.

While there is no method of preventing users from deleting rights objects, OMA requires that devices implement a simple replay protection mechanism that prevents users from deleting rights objects with exhausted stateful constraints and re-installing the original unexhausted rights object.

As we can see no way of implementing a retention criterion in OMA, and users are prevented from illegitimately restoring stateful rights objects by OMA's replay protection specification, we will consider all rights objects and associated state to be kept indefinitely.

6.5 Stateful Constraints

OMA devices support only unshared stateful constraints. In order to implement shared stateful constraints using OMA devices, we must assume that master rights issuers can issue domain controllers with *master rights objects* that permit domain controllers to maintain state information on behalf of the domains that they can control. Such master rights objects must be written in a language more powerful than OMA REL that we will not discuss here.

OMA REL supports three stateful constaints, called `count`, `timed-count` and `accumulated`. Shared versions of these constraints can be implemented by having the domain controller divide the state information from the master rights object into quanta that can be issued to domain members as normal rights objects:

Count. Every time a device wishes to exercise a right, the domain controller must issue it with a rights object containing a `count` constraint of one, and decrease the count remaining in the master rights object by one.

Timed-count. The `timed-count` constraint is similar to the `count` constraint, but allows the device to play the content for a specified time period without decrementing the counter. To implement a shared version, the domain controller may issue as many rights objects as it wishes containing an `accumulated` constraint equal to the timer value – that is, devices may play the content for the specified time period with restraint. Devices that want to play the content for longer periods must obtain a one-count rights object as for the `count` constraint.

Accumulated. Whenever a device wishes to exercise a right, the domain controller must issue it with a rights object containing an `accumulated` constraint with some value less than that remaining in the master rights object. The time remaining in the master rights object must be decremented by the amount of time awarded to the device.

More sophisticated approaches to shared stateful constraints are possible using more advanced protocols than those defined by OMA [14]. We cannot see any way of implementing such protocols using standard OMA devices, however.

7 Conclusion

There are many situations in which multimedia works are legitimately shared between a number of different users and/or multimedia terminals. Existing digital rights management systems, however, provide only very primitive support for sharing behaviours. We have proposed to marry the existing notions of an "authorised domain" and an "environment role" to provide flexible but controllable sharing of access to valuable digital content. Using sharing-oriented business models and concepts from ubiquitous computing, it may be possible to achieve content sharing that is virtually invisible except to those who would share without limit.

The power and flexibility of a system implemented in our model depends to a large extent on the language used for composing domain licences. We have proposed a domain expression language based on the Open Digital Rights Language that includes a number of useful domain criteria. Our language is a preliminary proposal, however, and we expect that it could be refined by further examination of business models and scenarios.

We have demonstrated the practicality of our model by outlining how it could be implemented using OMA domains and ubiquitous computing. Our model increases the expressive power of OMA's digital rights management specification without requiring end-user devices to be upgraded.

Using standard devices, however, has a number of drawbacks compared to what could be achieved using upgraded devices that support more sophisticated

approaches to key management and communication within a domain. We will leave the development of such approaches as future work.

8 Acknowledgements

This work was partially funded by the Smart Internet Technology Co-operative Research Centre, Australia. We would particularly like to thanks members of the Content Management Group at Telstra Research Laboratories and members of the User-Centred Design Group at the Royal Melbourne Institute of Technology for stimulating discussion in this area.

References

[1] J.-P. Andreaux, A. Durand, T. Furon, and E. Diehl. Copy protection system for digital home networks. *IEEE Signal Processing Magazine*, 21(2):100–108, 2004.

[2] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. GEO-RBAC: A spatially aware RBAC. In *ACM Symposium on Access Control Models and Technologies*, pages 29–37, 2005.

[3] B. Brown, A. J. Sellen, and E. Geelhoed. Music sharing as a computer supported collaborative application. In *European Conference on Computer-Supported Collaborative Work*, pages 179–198, 2001.

[4] G. Chen and D. Kotz. Supporting adaptive ubiquitous applications with the Solar system. Technical Report TR2001-397, Dartmouth College, USA, 2001.

[5] A. Corradi, R. Montanari, and D. Tibaldi. Context-based access control for ubiquitous service provisioning. In *International Computer Software and Applications Conference*, pages 444–451, 2004.

[6] M. J. Covington, W. Long, S. Srinivasan, A. K. Dey, M. Ahamad, and G. D. Abowd. Securing context-aware applications using environment roles. In *ACM Symposium on Access Control Models and Technologies*, pages 10–20, 2001.

[7] S. Guth, G. Neumann, and M. Strembeck. Experiences with enforcement of access rights extracted from ODRL-based digital contracts. In *ACM Digital Rights Management*, pages 90–102, Washington, DC, USA, 2003.

[8] W. Han, J. Zhang, and X. Yao. Context-sensitive access control model and implementation. In *International Conference on Computer and Information Technology*, pages 757–763, 2005.

[9] F. Hansen and V. Oleshchuk. Spatial role-based access control model for wireless networks. In *IEEE Vehicular Technology Conference*, pages 2093–2097, 2003.

[10] C. Hibbert. Copy protection work in the DVB. *DVB-Scene*, 05:14–15, 2002.

[11] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma. Context sensitive access control. In *ACM Symposium on Access Control Models and Technologies*, pages 111–119, 2005.

[12] International Standards Organisation. Information technology – multimedia framework (MPEG-21) – part 5: Rights expression language. ISO/IEC 21000-5:2004.

[13] P. Koster, F. Kamperman, P. Lenoir, and K. Vrielink. Identity based DRM: Personal entertainment domain. In *IFIP Conference on Communications and Multimedia Security*, pages 42–54, 2005.

[14] Q. Liu, R. Safavi-Naini, and N. P. Sheppard. Maintaining count constraints in a household domain in DRM. In *International Workshop on Software Support for Portable Storage*, pages 69–75, San Francisco, USA, 2005.

[15] Marlin Developer Community. Marlin – core system specification version 1.2. `http://www.marlin-community.com`, 12 April 2006.

[16] B. Marušič, P. de Cuetos, L. Piron, and Z. Lifshitz. TIRAMISU: That's unobtrusive DRM in the home domain. *Indicare Monitor*, 2(5), July 2005. `http://www.indicare.org/tiki-read_article.php?articleId=125`.

[17] C. Masone. Role Definition Language (RDL): A language to describe context-aware roles. Technical Report TR2002-426, Dartmouth College, USA, 2002.

[18] Microsoft Corporation. Windows Server 2003 Rights Management Services. `http://www.microsoft.com/windowsserver2003/technologies/rightsmgmt/defa%ult.mspx`, 2005.

[19] Open Digital Rights Language Initiative. The Open Digital Rights Language Initiative. `http://odrl.net`, 2004.

[20] Open Mobile Alliance. OMA DRM v2.0 approved enabler, 3 March 2006.

[21] F. Pestoni, J. B. Lotspiech, and S. Nusser. xCP: Peer-to-peer content protection. *IEEE Signal Processing Magazine*, 21(2):71–81, 2004.

[22] B. C. Popescu, B. Crispo, A. S. Tanenbaum, and F. L. A. J. Kamperman. A DRM security architecture for home networks. In *ACM Workshop on Digital Rights Management*, pages 1–10, 2004.

[23] K. H. Reddy, G. Lao, and A. Budo-Marek. Protected content distribution system. US Patent 6,824,051, 2004.

[24] G. Sampemane, P. Naldurg, and R. H. Campbell. Access control for active spaces. In *Annual Computer Security Applications Conference*, pages 343–352, 2002.

[25] Secure Video Processor Alliance. SVP open content protection system: Technical overview. `http://www.svpalliance.org/docs/e2e_technical_introduction.pdf`, 3 January 2005.

[26] S. Sovio, N. Asokan, and K. Nyberg. Defining authorization domains using virtual devices. In *Symposium on Applications and the Internet Workshops*, page 331, 2003.