

# Comparing Logic-based and XML-based Rights Expression Languages\*

Cheun Ngen Chong, Sandro Etalle, and Pieter H Hartel

Faculty of EEMCS  
University of Twente  
The Netherlands

{chong, etalle, pieter}@cs.utwente.nl

## Abstract

Several rights expression languages (RELs) have been proposed to describe licenses governing the terms and conditions of content access. In this field XrML and ODRL cover a prominent role. Both languages are powerful yet complex. In this paper we propose a way of analysing RELs and we apply it to ODRL, XrML and to LicenseScript, a REL we have proposed in [3]. In addition, we test these languages against a number of example scenarios. These examples bring new insights, and shed new light on some of the limits of XrML and ODRL.

## 1 Introduction

Right expression languages (RELs) are languages devised specifically to express condition of use of digital content in general, and of multimedia in particular. Right expression languages can for instance be used to describe an agreement between a content provider and an music distributor, or to express the copyright associated to a given piece of music, by specifying under which conditions the user is allowed to play, broadcast, or copy.

In the vast scene of multimedia delivery, two RELs in particular have attained a prominent position: XrML [5] and ODRL [6]. The eXtensible rights Markup Language (XrML) ([www.xrml.org](http://www.xrml.org)) is proposed and maintained by ContentGuard, Inc. ([www.contentguard.com](http://www.contentguard.com)), and has its roots in the Stefik's Digital Property Rights Language. XrML is adopted by Microsoft in Windows Media Player 9. The Open Digital Rights Language (ODRL) ([www.odrl.net](http://www.odrl.net)), on the other hand, was proposed by Renato Iannella from IPR Systems Ltd. ([www.iprsystems.com](http://www.iprsystems.com)), and is endorsed by the Open Mobile Alliance (OMA) ([www.openmobilealliance.org](http://www.openmobilealliance.org)).

XrML and ODRL have many similarities: syntactically they are both based on XML while structurally they both conform to the Stefik's axiomatic principles of rights modelling ([www.oasis-open.org/cover/DPRLmanual-XML2.html](http://www.oasis-open.org/cover/DPRLmanual-XML2.html)).

XML-based DRLs, however, have some intrinsic disadvantages: (1) the syntax is complicated and obscure when the conditions of use become complex, (2) these languages lack a formal semantics [9, 4] the meaning of licenses relies heavily on the human interpretation, and (3) the language cannot express many useful copyright laws [7]. To address these problems we have proposed a new, logic-based REL, named LicenseScript [3]. LicenseScript has a declarative as well a procedural reading (i.e., can be used as a programming language), and this makes it possible to capture a multitude of sophisticated usage patterns precisely and unambiguously.

LicenseScript provides an approach to REL which is diametrically opposite to that of XrML and ODRL: it is logic-based rather than XML-based. This makes it extremely difficult to make an objective assessment and comparison of the two REL styles. Such an objective assessment is of course of great importance for a clear understanding of the advantages and the limitation of XrML and ODRL. Last but not least, it must be noted that making such assessment is far from trivial, as ODRL's and XrML's specifications are amazingly long and complex.

This paper aims at at least partially solving this problem. Our contribution is twofold: first, we present a simple method for analysing the anatomy of a REL, and we apply it to ODRL, XrML and LicenseScript; secondly, we analyse in depth a number of examples, which we have coded in LicenseScript as well as in ODRL and XrML. In our opinion, these examples bring new insights, and shed new light on some of the weaknesses

---

\*This work is sponsored by Telematica Instituut, The Netherlands.

of XrML and ODRL.

The organization of the remainder paper: Section 2 discusses the anatomy of the RELs. Section 3 describes briefly the LicenseScript language to make the paper self-contained. Section 4.1 and section 4.2 translate the scenarios presented in ODRL and XrML specifications into LicenseScript, respectively. Section 4.3 describes some of the novel scenarios in LicenseScript. Lastly, section 5 concludes the paper and presents future work.

## 2 Anatomy of RELs

To aid the comparison of RELs we propose an anatomy of the RELs. Based on Stefik’s axiomatic principles, the XrML and ODRL specifications, and the requirements of RELs proposed by Parrott [8], we conclude that RELs have a structure which is shown in Figure 1. The figure is presented in the form of a class diagram because this exhibits the logical relations between the components. This figure provides an abstract view of a REL.

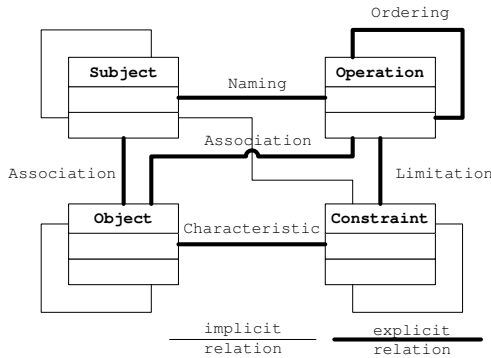


Figure 1: The components and their relations in a REL.

We identify four main components, namely *subject*, *object*, *operation*, and *constraint*. We explain these components in section 2.1. Each of these components is logically related to other components. We elaborate these *relations* in section 2.2. The components and the relations established within support a wide variety of *models* of the rights management systems. We elaborate the models in section 2.3.

Most of the follows come from the existing material (Parrott [8] and XML-based RELs specification), but we explicitly indicate the additional features of a REL.

### 2.1 Components

From Parrott’s requirements of RELs [8] and the XML-based RELs specification, we conclude that there are

four main components in a REL, namely (1) *subject*, which is an actor who performs some operations; (2) *object*, which is a content acted upon by a subject; (3) *operation*, which is what a subject can do to an object; and (4) *constraint*, which describes when an operation can be performed. There are two types of operation: a *right* is an operation that can be performed directly on the object; and an *obligation* is an operation that must precede or follow another operation.

As an illustration consider the following:

**Example 1** *Alice wants to play a high-quality movie for three times, she must pay \$5 upfront.*

Intuitively, the subject is “Alice”, the objects are “movie” and “\$5”, the right is “play”, the obligation is “pay”, and the constraints are “high-quality”, “for three times” and “upfront”.

**Subject** A REL typically describes the subject by naming, e.g. “Alice”. Additionally, the REL must be able to distinguish the type of the subject by using the names, e.g. creator, end-user, distributor and so on. A REL must also be able to specify the identification mechanism employed by common rights management systems to describe the subject, e.g. a digital certificate and public key . This is discussed in section 2.3.

**Object** From Parrott’s requirements of RELs [8], similar to subject, a REL uses names to describe an object, such as the title of the object or the artist. A REL is required to support (1) generalized types of objects, for instance, multimedia (e.g. MP3), personal data (e.g. DOC) or meta-data (e.g. XML); (2) classification of similar objects, for instance, “publisher Addison’s ebooks”; (3) fuzzy (or implicit) matching criteria of the object, for instance, “looks like” and “sounds like”; and (4) the delivery methods of the object, for example, by downloading, streaming or by means of physical storage (e.g. CD).

**Operation** As mentioned earlier, there are two types of operation: *right* and *obligation*. There are various types of rights identified by Rosenblatt et al [10]: (1) *Render*, which indicates a set of rights in which the object can be consumed, e.g. play; (2) *Reuse*, which indicates a set of rights in which the object can be re-utilized, e.g. modify. (3) *Transport*, which indicates a set of rights in which the subject’s rights over the object can be transferred, e.g. lend. (4) *Object management*, which indicates a set of rights to handle the management over the object, e.g. move and duplicate. None of these cover the regulation of the rights proper. Therefore, in addition, we propose a further set of rights,

namely (5) *Rights regulation*, which indicates a set of rights that regulate the subject’s rights over the object, e.g. update and renew. A REL should be capable of describing different obligations. An obligation may be an operation that enables or activates the rights over the objects. The *pay* and *register* operations are two common examples.

**Constraint** Parrott [8] and XML-based RELs specifications recognize several common constraints: (1) *temporal*, such as date and time (e.g. the ebook can be viewed before 20 March 2004); accumulated (e.g. the ebook can be viewed for 2 weeks); and interval (e.g. the ebook can be viewed within 20 days from the time of issuing this license); (2) *bound*, for instance, the number of distinct times the ebook can be viewed, and the range of the page numbers of the ebook that can be printed; (3) *environment*, which may be a physical environment (e.g. geographic territory) or logical environment (e.g. network address or system environment); (4) *aspect*, which mainly relates to the technical perspectives of the object, for example, quality and format of the content; (5) *purpose*, for instance, educational purpose and commercial reason. There may be more unique constraints required when new scenarios emerge.

We introduce another constraint, namely the *status* constraint. Real time content access requires this constraint to indicate the current state, e.g. availability and accessibility of the content at the time the rights are exercised.

Components		XrML	ODRL	LS
Right	Render	+	+	+
	Reuse	+	+	+
	Transport	+	+	+
	Manage Object	+	+	+
	Regulate Rights	o	o	+
Obligation		+	+	+
Object		+	+	+
Subject		+	+	+
Constraint	Temporal	+	+	+
	Bound	+	+	+
	Environment	+	+	+
	Aspect	+	+	+
	Purpose	o	+	+
	Status	o	o	+

Table 1: Comparisons of XrML, ODRL and LicenseScript from the perspective of components [‘+’(can), ‘o’(maybe), ‘-’(cannot)].

Table 1 shows that XrML and ODRL are able to represent render, reuse, transport, and object management rights. However, XrML and ODRL do not accommodate (explicitly) the descriptions of rights that regulate other rights. For example, “a user can *renew* the rights to play a movie within a fixed period (after the expiry

time of the rights) with a discount”. However, XrML and ODRL do cater for the revocation of rights and obligations.

XrML does not provide explicit facilities to specify the purpose constraints. ODRL and XrML cannot express the status of the object. However, LicenseScript is able to accommodate (all) the listed constraints.

## 2.2 Relations

A REL must specify *relations* between components. As can be seen in Figure 1, there are two distinct types of relations, namely *explicit* relations and *implicit* relations. We use example 1 to elaborate some of the relations discussed in this section.

Parrott [8] identifies two classes of explicit relations, namely *ordering* relation, e.g. “pay \$5 before play the movie” (operation–operation); and *association* relation, “Alice owns the movie” (subject–object) and “play is for movie” (operation–object). The *ordering* relation describes how operations are linked. For example, “pay before play” is an example of antecedent obligation; and “play then pay” is an example of consequent obligation. An ordering can be total or partial. A total ordering fully specifies the order of all operations, for example, “register, pay and then play”. A partial ordering implies there is no explicit order between all items, for example, “register and then play, user can pay before or after”. The *association* relation covers the subject–object and operation–object relations.

We identify three additional types of explicit relations, namely, (1) *naming* relation (subject–operation), which specifies the name of the operation the subject can perform, e.g. “Alice plays the music”; (2) *limitation* relation, which implies that the operations are restricted by the constraints, in the same example, (constraint–operation); and (3) *characteristic* relation (constraint–object), which describes the object (that the operations can be acted upon), e.g. “high-quality movie”.

We also identify several implicit relations (see Figure 1), which include: subject–subject, subject–constraint, object–object, and constraint–constraint. These implicit relations are embedded and indirect. To elaborate these relations, we use two additional examples:

**Example 2** Alice needs Bob to prove her identity so that she can play the movie.

**Example 3** Alice can reuse the image in the ebook to be used in her Web site, for educational purpose and for 2 years.

Example 2 exhibits the implicit subject–subject relations between “Alice” and “Bob”, as well as implicit subject–constraint between “Alice” and “prove

her identity”. Example 3 exhibits implicitly the object-object relations between the “image” and “ebook”, and the constraint-constraint relations between the “educational purpose” and “2 years”.

## 2.3 Models

A *model* describes a typical way of using a REL: (1) *revenue* model, (2) *provision* model, (3) *operational* model, (4) *contract* model, (5) *copyrights* model and (6) *security* model. A rights management system may exhibit different models simultaneously.

The *revenue* model, is normally related to the payment architecture of the system. There are myriad of revenue models, for example, pay-per-use, pay-upfront, pay-flatrate, tiered payment (e.g. free now pay later), pay to multi-entities (e.g. pay half to publisher and half to distributor), and fraction payment (e.g. discount and tax). New revenue models emerge every day.

The *provision* model may provide an *alternative solution* more than yes or no to the situations when the rights and obligations fail to meet the constraints. For instance, if viewing a high-resolution video is not allowed, it should be possible to switch to low-resolution video. Additionally, the provision model should be able to *reconcile the conflicts* caused, for example, when there is more than one subject performing the same operation on the same object simultaneously. The provision model also accommodates the *default settings of operations* over an object when the object is not associated with any operations. The *security* model defines a variety of security mechanisms, for instance, identification, authentication and authorization (IAA), access control, non-repudiation, integrity, audit trails and privacy.

The *operational* model handles the technological aspects of the system, such as quality-of-service, watermarking, caching, network operations, bandwidth and so forth. The *contract* model establishes the agreement of the terms and conditions (over the operations offered over the object and constraints) established between different subjects. We include the *copyrights* model in this category because the copyrights enforcement from the user’s standpoint is always a source of controversy [2]. The *copyrights* model enforces the copyrights acts (especially from the end-user’s standpoint), such as fair use, first sale and so on.

As can be seen in Table 2, all RELs are able to support the revenue models. However, XrML and ODRL are not able to support the provision model of reconciling the rights conflicts. This model handles the dynamic license evolutions and content access patterns. XrML and ODRL are static RELs that are not sufficiently flexible to meet this requirement.

Models		XrML	ODRL	LS
Revenue Model		+	+	+
Provision Model	Conflicts	o	o	+
	Alternatives	+	+	+
	Defaults	+	+	+
Operational Model		+	+	+
Contract Model		+	+	+
Security Model		+	+	+
Copyrights Model		o	o	o

Table 2: Comparisons of XrML, ODRL and LicenseScript from the perspective of Models.

None of the RELs can as yet support the copyrights model [11]. However, Mulligan and Burstein [7] provides several suggestions to incorporate the copyrights into the XML-based RELs. We address this issue as our future work.

In the following section, we describe concisely the LicenseScript language using a simple scenario as an example.

## 3 LicenseScript Language

LicenseScript is a language that is based on (1) multiset rewriting, which captures the dynamic evolution of licenses; and (2) logic programming, which captures the static terms and conditions on a license. LicenseScript provides a judicious choice of the interfacing mechanism between the static and dynamic domains.

A license specifies when certain operations on the object are permitted or denied. The *license* is associated with the *content*, as can be seen in Figure 2. A license carries *bindings*, which describe the attributes of the license; and the *clauses*, which determine if a certain operation is allowed (or forbidden). The license clauses consult the license bindings for their decision making and may also alter the values of the license bindings.

Licenses are bound to the terms that reside in multisets. For the specification of the licenses, we use logic programming. The readers are thus assumed to be familiar with the terminology and the basic results of the semantics of logic programs.

Figure 2 illustrates that ① an *operation* (performed by a subject) ② invokes a *rule* in the multiset. The rule then generates and executes a ③ *query* on the license ④ clauses. The ⑤ execution result of the rule is a newly generated license. We elaborate this transformation process later by using a simple scenario.

Now we use a simple illustrative scenario to explain the LicenseScript:

**Example 4** *Amanda gets an ebook, titled “King\_James\_A\_Book” from Ben Publisher. Ben issues a license with an expiry date fixed at “23/06/2004”.*

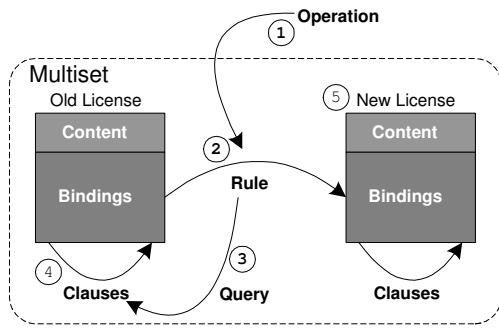


Figure 2: Transformation of licenses with content and bindings caused by rules.

This license allows Amanda to view the ebook and print two copies of the ebook:

```

license(ebook-King_James_A_Book,      L01
[   (canview(B1,B2,User) :-           L02
    get_value(B1,consumer,C),         L03
    C = User,                          L04
    get_value(B1,expires,Exp),        L05
    today(D),                          L06
    Exp > D),                          L07
  (canprint(B1,B2,User) :-           L08
    get_value(B1,consumer,C),         L09
    C = User,                          L10
    get_value(B1,printed,P),          L11
    get_value(B1,max_prints,Max),     L12
    P <= Max,                          L13
    set_value(B1,printed,P+1,B2)],    L14
[   (company=ben_publisher),          L15
    (consumer=amanda),                 L16
    (expires=23/06/2004),              L17
    (max_prints=2), (printed=0)])     L18

```

(L01,...,L18 are the line numbers of the code for the benefits of the readers. They are not part of the code.)

The license is represented by a term in the form of `license(content, C, B)`, where `content` is a unique identifier referring to the real content (`ebook-King_James_A_Book`); `C` is a list of *license clauses* consisting of Prolog programs describing when operations are permitted or denied; and `B` is a list of *license bindings* capturing the attributes of the license.

We define two rewrite *rules*, as shown below, to model the interface between the system and the licenses. The syntax of the rules is based on the Gamma notation [1] of multiset rewriting:

```

view(Ebook,User)  :           R01
  license(Ebook,C,B1) ->      R02
  license(Ebook,C,B2)         R03
  <= C |- canview(B1,B2,User)  R04
print(Ebook,User) :           R05
  license(Ebook,C,B1) ->      R06
  license(Ebook,C,B2)         R07
  <= C |- canprint(B1,B2,User) R08

```

(R01,...,R08 are the line numbers of the code for the benefits of the readers. They are not part of the code.)

We will step through the example assuming Amanda would like to view the eBook with the available licenses on her system:

1. Amanda's system wants to know whether she has the view *right* on the ebook. This is achieved by applying the view *rule* (R01) with appropriate parameters: `view(ebook-King_James_A_Book, amanda)`.
2. The rule finds the `license(ebook-King_James_A_Book, [...], [...])` (R02, L01) in the system. The first list refers to the license clauses (L02–L14), while the second list refers to the set of license bindings (L15–L18).
3. The rule then executes a *query* in the form of `canview(B1, B2, User)` (R04), where `B2` designates the output generated by the query (which is the new set of license bindings).
4. The license interpreter retrieves the value of the license binding `consumer` from the list of license bindings `B1` (L03) and compares the retrieved value with the user identity `User` (L04). `User` is passed in as an argument to the license clause (L02).
5. Similarly, the interpreter retrieves the value of the binding `expires` (L05).
6. The interpreter calls the primitive (which is discussed later) `today(D)` (L06) to obtain the current time and date.
7. The expiry date of the license must be greater than current time and date (L07).
8. If all conditions are satisfied (the user is valid and the license has not expired), the query returns `yes` (R04) to the interpreter, with the newly generated license bindings in `B2`. In this case the binding list is preserved.
9. The value `yes` indicates that the execution of `canview(B1, B2, User)` yields *success* in the license clauses `C`.
10. The rule `view(Ebook, User)` generates a new license with the newly generated license bindings `license(Ebook, C, B2)` (R03). In this case, as the license bindings are preserved, the newly generated license is the same as the old license.

Similarly, to decide whether Amanda has the *print* right, the print rule is applied. Here, new license bindings are generated after the execution of the rule: the value of the license binding `printed` is incremented (L14) every time the print operation succeeds.

There are a number of primitives to model the interface of the system with the license (interpreter), as shown in Table 3. We use these primitives in our licenses.

Primitive	Description
<code>get_value (B, n, V)</code>	Report in <i>V</i> the value of <i>n</i> from <i>B</i> .
<code>set_value (B, n, V, B')</code>	Give value <i>V</i> to <i>n</i> from <i>B</i> to <i>B'</i> .
<code>today (D)</code>	Bind <i>D</i> to the current system date.
<code>identify (ID)</code>	Retrieve the current domain identity to <i>ID</i> .
<code>transfer (P, M)</code>	Transfer of money <i>M</i> to entity <i>P</i> .

Table 3: Primitives used to model the interface between the system and the licenses.

For further details of the LicenseScript language, we may refer to References [3].

## 4 Scenarios

In this section, we discuss several scenarios from the ODRL specification and translate the scenarios into LicenseScript to demonstrate its flexibility and power.

The scenarios presented in the ODRL specification include ebook scenarios #1, #2 and #3, a video scenario, a superdistribution scenario, a software scenario, an image scenario and an audio scenario. We have translated these scenarios into LicenseScript (see Appendix A).

We also discuss some of the scenarios specified in XrML Example Use Cases ([www.xrml.org/spec/2001/11/ExampleUseCases.htm](http://www.xrml.org/spec/2001/11/ExampleUseCases.htm)): unlimited/limited usage, preview/promotional model, tiered pricing model, pay per view, subscription, territory restriction, temporal ordering of rights, usage of part of a work, payment according to different users, site license, pay multi-entities, personal lending and giving, superdistribution, unrestricted sales, personal copies, fraction payments, web service access, software execution, confidentiality of rights, operational model and secure device. We have also translated these scenarios into LicenseScript, as shown in Appendix B.

We also quote some of the text descriptions of the scenarios from the ODRL and XrML specifications in the following sections. In subsequent sections, we use LicenseScript to describe some of the scenarios presented in the XrML and ODRL.

### 4.1 ODRL Scenarios

For demonstration purpose, in this section, we only chose two of the aforesaid scenarios, namely Ebook Scenario #1, #2 and #3, as described in section 4.1.1 to section 4.1.3.

#### 4.1.1 Ebook Scenario #1

An author, named Corky Rossi and an illustrator, named Addison Rossi publish their ebook via “EbooksRUS Publishers”. The publishers wish to offer consumers the opportunity to purchase (for \$AUD20.00 plus 10% tax) the ebook which is restricted to a single CPU only. Buyers are allowed to print a maximum of 2 copies. They will also allow the first 5 pages (Units) of the ebook to be viewed online for free (no requirement). The revenue split is 60% to the author, 10% to the illustrator and 30% to the publisher.

This scenario precedes the contract model. In other words, an offer (which is an unsigned or unagreed contract) is involved in this scenario. The offer for this scenario can be represented in LicenseScript as follows:

```
offer(urn:ebook.world/999999/ebook/rossi-000001,
[ (candisplay (B,B,P1,P2) :-
  P1>=1, P2>5, identify (Id1),
  get_value (B,cpu_id,Id1),
  Id1=Id2, paid=true),
(candisplay (B,B,P1,P2) :-
  P1>=1, P2<=5),
(canprint (B1,B2) :-
  get_value (B1,paid,Paid), Paid=true,
  get_value (B1,prints,Prints), Prints<2,
  Newprints is Prints+1,
  set_value (B1,prints,Newprints,B2)),
(canpay (B1,B2,ToA,ToI,ToP) :-
  get_value (B1,paid,Paid), Paid=false,
  get_value (B1,rate,Rate),
  get_value (B1,tax_percent,Tax),
  Price is Rate*(1+Tax),
  ToA is Price*0.6,
  ToI is Price*0.1,
  ToP is Price*0.3,
  set_value (B1,paid,true,B2)),
(canagree (B1,B2,C,CPU) :-
  get_value (B1,consumer,Consumer),
  Consumer=nil, set_value (B1,consumer,C,B2),
  set_value (B1,cpu_id,CPU,B2),
  set_value (B1,prints,0,B2),
  set_value (B1,paid,false,B2)
]),
[ (currency=aud), (rate=20.00),
(tax_percent=10.00), (cpu_id=nil),
(paid=false), (prints=0),
(author="x500:c=AU;o=RightsDir;cn=CorkyRossi"),
(illustrator="x500:c=AU;o=RightsDir;cn=AddisonRossi"),
(publish="x500:c=AU;o=RightsDir;cn=EBooksRUS"),
(consumer=nil)]])
```

The arguments `P1` and `P2` (in `candisplay`) designate the starting page and ending page, respectively; `ToA`, `ToI`, `ToP` (in `canpay`) indicate the amount to be paid to the author, illustrator, and publisher, respectively; `C` and `CPU` (in `canagree`) refer to the consumer identity and the CPU (number) the consumer is using.

To represent offer, the license bindings `consumer` and `cpu_id` are `nil`. The license (agreement) is generated from this offer, the moment when the buyers agree this offer. We model this process by introducing a new rewrite rule `agree`:

```
agree (Ebook, C, CPU) :
  offer (Ebook, C, B1) ->
  offer (Ebook, C, B1), license (Ebook, C, B2)
<= C |- canagree (B1, B2, C, CPU)
```

When the rule `agree` is executed, a new license (`license (Ebook, C, B2)`) representing the agreement determined is generated and assigned to the consumer. At the same time, the offer from the previous scenario `offer (Ebook, C, B1)` is preserved after the rule execution.

The agreement model (contract model) is rendered explicit in LicenseScript (by using the rule `agree` and the license clause `canagree`). LicenseScript is able to express the transition from offer to agreement explicitly.

The next section illustrates an example of a license generated from this offer, through the rule of `agree`.

#### 4.1.2 Ebook Scenario #2

Following the Ebook Scenario #1, a consumer (Mary Smith) purchases the ebook under the conditions outlined. The agreement from Ebook Scenario #1 now has a context (with identifier and date). The permission now shows the details of the constraint that is specific for the consumer (in this case, the CPU identifier is saved within the license).

In other words, the license for this scenario (as shown below) represents an agreement document established between the consumer and the content provider. The license generated from the agreement can be represented as follows:

```
license (urn:ebook.world/999999/ebook/rossi-000001,
[ (candisplay (B, B, P1, P2) :-
  P1>=1, P2>5, identify (Id1),
  get_value (B, cpu_id, Id2),
  Id1=Id2, paid=true),
(candisplay (B, B, P1, P2) :-
  P1>=1, P2<=5),
(canprint (B1, B2) :-
  get_value (B1, paid, Paid), Paid=true,
  get_value (B1, prints, Prints), Prints<2,
  Newprints is Prints+1,
  set_value (B1, prints, Newprints, B2)),
(canpay (B1, B2, ToA, ToI, ToP) :-
  get_value (B1, paid, Paid), Paid=false,
  get_value (B1, rate, Rate),
  get_value (B1, tax_percent, Tax),
  Price is Rate*(1+Tax),
  ToA is Price*0.6,
  ToI is Price*0.1,
  ToP is Price*0.3,
  set_value (B1, paid, true, B2)),
(canagree (B1, B2, C, CPU) :-
  get_value (B1, consumer, Consumer),
  B1 = nil, set_value (B1, consumer, C, B2),
  set_value (B1, cpu_id, CPU, B2),
  set_value (B1, prints, 0, B2),
```

```
set_value (B1, paid, false, B2)
]),
[ (currency=aud), (rate=20.00),
(tax_percent=10.00), (paid=false), (prints=0),
(cpu_id="Adobe-WebBuy:CPD-ID:ER-393939-DSS-787878"),
(author="x500:c=AU;o=RightsDir;cn=CorkyRossi"),
(illustrator="x500:c=AU;o=RightsDir;cn=AddisonRossi"),
(publish="x500:c=AU;o=RightsDir;cn=EBooksRUS"),
(consumer="urn:ebook.world/99999/users/msmth-00011")]]
```

The clauses `candisplay`, `canprint`, and `canpay` correspond to the operations `display`, `print` and `pay` stated in the scenario. The clause `canagree` is a license clause we use to model the agreement, which we discuss later in Ebook Scenario #2.

To model the payment method, we introduce a *wallet*. The wallet, the offer and the license are elements of the multiset. The wallet is represented as a term of the form `wallet (Cw, Bw)`, where `Cw` is a Prolog program, and `Bw` is the set of bindings. Similar to the license (as described in section 3), in the wallet we have clauses that allow rules to perform operations over the wallet. We assume that one binding named `money` is always in `Bw`. This designates the amount of the money in the wallet.

The wallet contains two clauses, namely `canload` and `cantransfer`, which enables the user to increase the balance of the wallet and to transfer the money to other entities, respectively. The wallet is shown as follows:

```
wallet (
[ (canload (Bw1, Bw2, A) :-
  get_value (Bw1, money, Money),
  N is A+Money,
  set_value (Bw1, money, N, Bw2)),
(cantransfer (Bw1, Bw2, ToA, ToI, ToP) :-
  get_value (Bw1, money, Money),
  Money >= ToA+ToI+ToP,
  get_value (Bw1, author, A),
  get_value (Bw1, illustrator, I),
  get_value (Bw1, publish, P),
  transfer (A, ToA), transfer (I, ToI),
  transfer (P, ToP),
  N is Money-(ToA+ToI+ToP),
  set_value (Bw1, money, N, Bw2)) ],
[ (money=x) ])
```

Now, we can define three rules for operations *display*, *print* and *pay*, as follows:

```
display (Ebook) :
  license (Ebook, C, B) -> license (Ebook, C, B)
<= C |- candisplay (B, B)
print (Ebook) :
  license (Ebook, C, B1) -> license (Ebook, C, B2)
<= C |- canprint (B1, B2)
pay (Ebook) :
  license (Ebook, C, B1), wallet (Cw, Bw1) ->
  license (Ebook, C, B2), wallet (Cw, Bw2)
<= C |- canpay (B1, B2, ToA, ToI, ToP),
  Cw |- cantransfer (Bw1, Bw2, ToA, ToI, ToP)
```

Similar to the agreement model (contract model), the payment is rendered explicit.

From the study of these two scenarios, we conclude that LicenseScript is able to model a wide variety of distinct terms in a multiset, which could be a license, an offer, a wallet, etc. This *manifests a high flexibility in designing systems that are not confined to rights management*.

### 4.1.3 Ebook Scenario #3

We consider an Electronic Book Exchange voucher for an ebook entitled “XML: A Managers Guide”. The rights owner is Addison-Wesley. There is an agreement with the Distributor of this book (a company called “XYZ”). They have rights to Sell up to 5000 copies of the book.

A licensed end user for this book is “John Doe”. All his permissions start from the beginning of 2001 and expire at the end of 2004. He has rights to view the book for a total of 30 days. He can print up to 5 copies on a “trusted printer”. He can print up to 5 pages between page 1 and 100 every week - up to a total of 100 pages - on any printer. He can also extract 5000 bytes every week up to a total of 1,000,000 bytes onto the Clipboard (memory). He has the right to Give the book away after one year of the permissions starting.

The license for this scenario can be encoded in LicenseScript, as shown as follows:

```
license(urn:ebook.world/999999/ebook/rossi-000001,
[ (cansell(B1,B2,B3,D,U) :-
  get_value(B1,dist,Dist), Dist=D,
  get_value(B1,max_sells,MSells),
  set_value(B1,max_sells,MSells-1,B2),
  set_value(B1,user,U,B3),
  set_value(B1,max_sells,0,B3)),
(candisplay(B1,B2) :-
  get_value(B1,start,Start),
  get_value(B1,end,End),
  today(Today), Today>=Start, Today<=End,
  get_value(B1,display_days,Accdays),
  Accdays<=30,
  set_value(B1,display_days,Accdays+1,B2)),
(canprint(B1,B2,P,S,E) :-
  get_value(B1,start,Start),
  get_value(B1,end,End), today(Today),
  Today>=Start, Today<=End,
  get_value(B1,printer,Printer),
  P=Printer,
  get_value(B1,print_copies,Pcopies),
  Pcopies<=5),
(canprint(B1,B2,P,S,E) :-
  get_value(B1,start,Start),
  get_value(B1,end,End),
  today(Today), Today>=Start, Today<=End,
  S>=1, E<=100, Pages is E-S+1, Pages<=5,
  get_value(B1,prevprint_day,Prevprint),
  today(Today), Today<=Prevprint+7,
  get_value(B1,print_pages,Ppages),
  Ppages<=100, Newpages is Ppages+Pages,
  set_value(B1,print_pages,Newpages,B2),
  set_value(B1,prevprint_day,Today,B2)),
(canexcerpt(B1,B2,Bs) :-
  get_value(B1,start,Start),
  get_value(B1,end,End),
  today(Today), Today>=Start, Today<=End,
  get_value(B1,excerpt_maxbytes,Mbytes),
  Maxbytes+Bs<=1000000,
  get_value(B1,prevexcerpt_day,Prevdate),
```

```
today(Today), Today<=Prevdate+7,
get_value(B1,excerpt_weekbytes,Wbytes),
Wbytes+Bytes<=5000,
set_value(B1,excerpt_maxbytes,Mbytes+Bs,B2),
set_value(B1,excerpt_weekbytes,Wbytes+Bs,B2)),
(cangive(B1,B2,U) :-
  today(Today), Today>=give_date,
  set_value(B1,user,U,B2))
],
[ (license_issued_date=2001-05-01T08:30:00),
(give_date=2002-01-01T00:00:00),
(start=2001-01-01T00:00:00),
(end=2004-13-31T23:59:59),
(publish="http://publishers.net/registry/AWL"),
(dist="http://distributors.net/registry/xyz"),
(user="http://people.net/registry/john-doe-9999"),
(printer="guid:TrustPrint/4747474742222"),
(display_days=0), (max_sells=5000),
(print_copies=0), (print_pages=0),
(prevprint_day=0), (print_days=0),
(excerpt_days=0), (excerpt_counts=0),
(excerpt_maxbytes=0), (excerpt_weekbytes=0),
(prevexcerpt_day=0) ]]
```

The arguments D and U (clause cansell) indicate the distributor identity and user identity respectively; arguments S and E (clause canprint) indicate the start page and end page for printing, respectively; argument P in canprint indicates the printer identity; argument Bs indicate the number of bytes being extracted.

The corresponding rules for this scenario are:

```
sell(Ebook,D,B) :
  license(Ebook,C,B1) ->
  license(Ebook,C,B2), license(Ebook,C,B3)
  <= C |- cansell(B1,B2,B3,D,B)
display(Ebook) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- candisplay(B1,B2)
print(Ebook,P,S,E) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- canprint(B1,B2,P,S,E)
excerpt(Ebook,Bs) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- canexcerpt(B1,B2,Bs)
give(Ebook,U) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- cangive(B1,B2,U)
```

The generation of a new license from an old license (the agreement) is caused by the rule sell, when the content provider sells a copy of this ebook to a consumer. The content of the new agreement is altered (the number of copies that can be sold is decremented), as illustrated in license clause cansell. The license generated and assigned to the consumer prevents the consumer to sell this ebook to another entity (the number of max\_sells is set to 0).

These scenarios demonstrate that the license transformations caused by the operations are rendered distinct and explicit. These dynamic features and behaviours of the license are captured by the license clauses and the rules.

In the following sections, we do the same study on XML scenarios and translate them to LicenseScript.



## 4.2 XrML Scenarios

Like the ODRL scenarios, most of the XrML scenarios focus on the value chain from the content providers to the content end-users, except three, i.e. personal giving, personal lending, and superdistribution. We concentrate on the latter two.

### 4.2.1 Personal Lending

This scenario describes how a consumer can lend a movie to his/her friends during a rental period. This scenario illustrates how to specify the transfer of content and the rights, conditions, and obligations associated with it from one party to another, either temporarily or permanently. In this scenario, once a work is lent to someone else, the original user (as common sense) does not have access to the work until it is returned. The license can be written in LicenseScript as follows:

```
license(http://sonyPictures.com/AirForceOne,
[ (canplay(B,B,P) :-
  get_value(B,key_holder,OwnerKey),
  get_value(B,app_skey,ApproveKey),
  uddi_authorize(P,OwnerKey,ApproveKey),
  get_value(B,loaner,Loaner), Loaner=noone),
  (canplay(B,B,P) :-
  get_value(B,loaner,Loaner),
  Loaner=P, get_value(B,loan_start,LS),
  get_value(B,loan_end,LE),
  today(D), D >= LS, D <= LE),
  (canloan(B1,B2,O,L,S,E) :-
  get_value(B1,key_holder,OwnerKey),
  get_value(B1,app_skey,ServiceKey),
  uddi_authorize(O,OwnerKey,ServiceKey),
  get_value(B1,loaner,Loaner), Loaner=noone,
  get_value(B1,rental_start,RS),
  get_value(B1,rental_end,RE),
  today(D), D >= RS, D <= RE,
  set_value(B1,loan_start,S,B2),
  set_value(B1,loan_end,E,B2),
  set_value(B1,loaner,L,B2),
  uddi_track(P,L,TrackKey,LS,LE)),
  (canreturn(B1,B2,L) :-
  get_value(B1,loaner,Loaner),
  Loaner=L, set_value(B1,loaner,noone,B2),
  set_value(B1,loan_start,nil,B2),
  set_value(B1,loan_end,nil,B2))
],
[ (key_holder=<mod>Efgao6Nyf...<exp>AQAQAA=="),
  (app_skey="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
  (track_skey="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
  (loaner=noone), (rental_start="2001-05-20T19:28:00"),
  (rental_end="2001-05-29T19:28:00"), (loan_start=nil),
  (loan_end=nil)
])
```

The arguments of the clauses, P (in `canplay`), O, L, S, and E (in `canloan`) designate player value (i.e. the identification value), owner value, loaner value, start of time, and end of time for loan, respectively.

As can be seen from the license above, when the owner plays the video, the license clause `canplay` checks if this video is loaned to anyone (`Loaner=noone`). When the video is returned, the value of the bindings `loaner`, `loan_start`, and `loan_end` are reset.

The primitive `uddi_authorize` models the authorization via Universal Description, Discovery and Integration (UDDI) ([www.uddi.org](http://www.uddi.org)). Similarly, the primitive `uddi_track` models the report tracking via UDDI. We define the rules as follows:

```
play(Video,P) :
  license(Video,C,B) -> license(Video,C,B)
  <= C |- canplay(B,B,P)
loan(Video,O,L,S,E) :
  license(Video,C,B1) -> license(Video,C,B2)
  <= C |- canloan(B1,B2,O,L,S,E)
return(Video,L) :
  license(Video,C,B1) -> license(Video,C,B2)
  <= C |- canreturn(B1,B2,L)
```

Similar to the ODRL scenarios describe in section 4.1, the payment methods in this scenario are explicit. Additionally, the user authorization and the audit trailing of this scenario are implemented in the license clauses, with the primitives `uddi_authorize` and `uddi_track`, respectively.

### 4.2.2 Superdistribution

Alice can play an ebook for an up-front fee of \$2.00 and she can print it, as many times as she likes, for an up-front fee of \$3.00. Anyone can superdistribute this ebook.

This scenario shows the specification of content superdistribution in terms of associating the same or different sets of rights, conditions and obligations to the superdistributed content. The original consumer retains the rights granted to her, and new consumers are required to acquire the their own rights:

```
license(http://www.contentguard.com/xrmlUnleashed.spd,
[ (canplay(B,B) :-
  get_value(B,paid_play,Paid), Paid=true),
  (canprint(B,B) :-
  get_value(B,paid_print,Paid), Paid=true),
  (canextract(B,B) :-
  get_value(B,paid_extract,Paid), Paid=true),
  (canpay(B1,B2,T,P,I,A,O,Skey) :-
  T=play, get_value(B1,paid_play,Paid),
  Paid=false, get_value(B1,play_rate,A),
  get_value(B1,account,P),
  get_value(B1,institution,I),
  get_value(B1,pay_skey,Skey),
  set_value(B1,paid_play,true,B2)),
  (canpay(B1,B2,T,P,I,A,O,Skey) :-
  O = alice, T=print,
  get_value(B1,paid_print,Paid),
  Paid=false, get_value(B1,print_rate2,A),
  get_value(B1,account,P)
  get_value(B1,institution,I),
  get_value(B1,pay_skey,Skey),
  set_value(B1,paid_play,true,B2)),
  (canpay(B1,B2,T,P,I,A,O,Skey) :-
  O = anyone, T=print,
  get_value(B1,paid_print,Paid),
  Paid=false, get_value(B1,print_rate1,A),
  get_value(B1,account,P)
  get_value(B1,institution,I),
  get_value(B1,pay_skey,Skey),
  set_value(B1,paid_play,true,B2)),
  (canpay(B1,B2,T,P,I,A,O,Skey) :-
  T=extract, get_value(B1,paid_extract,Paid),
```

```

    Paid=false, get_value(B1,extract_rate,A),
    get_value(B1,account,P)
    get_value(B1,institution,I),
    get_value(B1,pay_skey,Skey),
    set_value(B1,paid_play,true,B2)),
  (canissue(B1,B2) :-
    set_value(B1,owner,anyone,B2))
],
[
  (owner=alice), (key_holder=<modulus>...<exp>...),
  (currency=USD), (paid_play=false),
  (play_rate=2.00), (paid_print=false),
  (print_rate1=15.00), (print_rate2=3.00),
  (paid_extract=false), (extract_rate=8.00),
  (pay_skey=D04951E4-332C-4693-B7DB-D3D1D1C20844),
  (institution=2223718664), (account=123457)
]
)

```

The arguments in `canpay`: `T` refers to the type of the payment (the purpose of the payment); `P` refers to account number of the party who makes the payment; `I` indicates the institution (bank); `A` indicates the amount of the payment; `O` indicates the owner identity; and `Skey` indicates the key value to make the payment transfer. The license clause `canissue` enables the user to distribute this license to `anyone`, which is the core feature of superdistribution.

Now we define several rules to achieve superdistribution as follows:

```

play(Ebook) :
  license(Ebook,C,B) -> license(Ebook,C,B)
  <= C |- canplay(B,B)
print(Ebook) :
  license(Ebook,C,B) -> license(Ebook,C,B)
  <= C |- canprint(B,B)
extract(Ebook) :
  license(Ebook,C,B1) ->
  license(Ebook,C,B1),license(Ebook,C,B2)
  <= C |- canextract(B1,B2)
issue(Ebook) :
  license(Ebook,C,B1) ->
  license(Ebook,C,B1),license(Ebook,C,B2)
  <= C |- canissue(B1,B2)
pay(Ebook,O,T) :
  license(Ebook,C,B1),wallet(Cw,Bw1) ->
  license(Ebook,C,B2),wallet(Cw,Bw2)
  <= C |- canpay(B1,B2,T,P,I,A,O,Skey),
  Cw |- cantransfer(Bw1,Bw2,I,P,Skey,A)

```

Again, we model the payment by using a `wallet`, as follows:

```

wallet(
[
  (canload(C1,C2,Amnt) :-
    get_value(C1,money,Money),
    N is Amnt+Money,
    set_value(C1,money,N,C2)),
  (canttransfer(C1,C2,Int,Acc,SKey,Amnt) :-
    get_value(C1,money,Money),
    Money >= Amnt,
    uddi_transfer(Acc,Int,Amnt,Skey),
    Newval is Money-Amnt,
    set_value(C1,money,Newval,C2))
],
[
  (money=x) ]
)

```

Analogous to the preceding scenarios, many aspects of XrML are made explicit in LicenseScript (e.g. payment, user authorization, and audit tracking). There

are different versions of license clauses `canplay` and `canprint` corresponding to various constraints, such as different price for the owners and other users who own the copies via superdistribution.

As we have seen in the preceding sections, LicenseScript is able to capture the scenarios of XML-based RELs presented in each specification. We have tried all examples from the ODRL and XrML specifications and have not found any that LicenseScript cannot express succinctly. Thus, we may conclude that the LicenseScript language is able to provide a fine granularity of control over the content, as XML-based RELs. In the following section, we illustrate novel scenarios that we encountered in other sources.

## 4.3 Novel Scenarios

Novel scenarios emerge everyday. The scenarios we introduce in this section are only the tip of the iceberg. We use some to demonstrate the advantages of LicenseScript over XML-based RELs.

These scenarios include “Project Document Sharing”, which manages project documents sharing and controls the versions of the documents and “License Evolution Modelling”, which is a scenario that utilizes the abstraction potential of LicenseScript. We capture these scenarios in LicenseScript.

### 4.3.1 Project Document Sharing

Fred, his project teammate Greg, and his project leader Han use a document sharing system for their project document sharing. Sometimes conflicts happen, for example, when Fred and Han are working on the same document at the same time. To reconcile these conflicts, they agree on a policy that Han’s version always overwrites others’ versions. Ian (from outside of the institute) has joined the project recently. Han has the privilege to grant him the system access.

The license for document “State-of-the-Art.tex”, which is created by Fred, can be written as follow:

```

license(tex:state-of-the-art.tex,
[
  (can_startread(B1,B2,U) :-
    get_value(B1,members,Ms), member(U,Ms),
    get_value(B1,systems,SYSSs),
    identify(D), member(D,SYSSs),
    get_value(B1,isreading,Rs),
    not(member(U,Rs)), append(U,Rs),
    set_value(B1,isreading,Rs,B2),
    get_value(B1,history,H),
    today(D), append([U,startread,D],H,H2),
    set_value(B1,history,H2,B2)),
  (can_endread(B1,B2,U) :-
    get_value(B1,isreading,Rs),
    remove(Rs,U,NR),
    set_value(B1,isreading,NR,B2),
    get_value(B1,history,H),
    today(D), append([U,endread,D],H,H2),
    set_value(B1,history,H2,B2)),
  (can_startwrite(B1,B2,U) :-

```

```

get_value(B1, members, Ms), member(U, Ms),
get_value(B1, systems, SYSs),
identify(D), member(D, SYSs),
get_value(B1, iswriting, Rs), length(Rs)=0,
append(U, Rs), set_value(B1, iswriting, Rs, B2),
get_value(B1, history, H),
today(D), append([U, startwrite, D], H, H2),
set_value(B1, history, H2, B2)),
(can_startwrite(B1, B2, U) :-
get_value(B1, members, Ms), member(U, Ms),
get_value(B1, systems, SYSs),
identify(D), member(D, SYSs),
get_value(B1, iswriting, Rs), length(Rs)>1,
get_value(B1, leader, L), U=L,
remove(X, iswriting, NL), append(U, NL, NL2),
set_value(B1, iswriting, NL2, B2)),
(can_endwrite(B1, B2, U) :-
get_value(B1, history, H),
today(D), append([U, endwrite, D], H, H2),
set_value(B1, history, H2, B2),
get_value(B1, iswriting, Rs), member(U, Rs)),
(cangrant(B1, B2, U1, U2, S) :-
get_value(B1, leader, L), U1=L,
get_value(B1, members, Ms), append(Ms, U2),
set_value(B1, members, Ms, B2),
get_value(B1, systems, SYSs),
identify(D1), member(D1, SYSs),
not(member(S, SYSs)), append(S, SYSs),
set_value(B1, systems, SYSs, B2))
],
[ (creator=fred), (leader=han),
(members=[fred, han, greg]),
(systems=[univ_twente]),
(isreading=[]), (iswriting=[]),
(history=[])
])

```

The argument *U* in all license clauses indicates the user's identity. The argument *S* in license clause `cangrant` designates the system identity of another user.

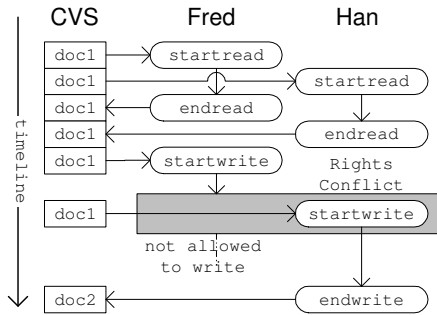


Figure 3: A timing diagram of rights (real-time) concurrent activities in the document sharing system.

The binding `members` is a list of members who have rights on this document. The binding `systems` is a list of system environments that are permitted for the members to access the document. This binding prevents the document from being accessed from an untrusted environment. The license bindings `isreading` and `iswriting` are two sets that indicate the users who are reading and writing the document currently, respectively. In other words, they indicate the current status of the document. The license binding `history` func-

tions as audit trail that records operations that have been performed by the users on the document.

There are 5 rules involved in this scenario, as can be seen as follows:

```

startread(Doc, U) :
    license(Doc, C, B1) -> license(Doc, C, B2)
    <= C |- can_startread(B1, B2, U)
startwrite(Doc, U) :
    license(Doc, C, B1) -> license(Doc, C, B2)
    <= C |- can_startwrite(B1, B2, U)
startwrite(Doc, U) :
    license(Doc, C, B1) -> license(Doc, C, B2)
    <= C |- can_startwrite(B1, B2, U)
endwrite(Doc, U) :
    license(Doc, C, B1) -> license(Doc, C, B2)
    <= C |- can_endwrite(B1, B2, U)
grant(Doc, U1, U2, D) :
    license(Doc, C, B1) -> license(Doc, C, B2)
    <= C |- cangrant(B1, B2, U1, U2, D)

```

This scenario exhibits a concurrent content access pattern: More than one subject may be performing the same operation on the same content simultaneously. This evokes conflicts. For instance, Figure 3 shows that when Fred and Han attempt to write at the same time, the project policy prevails, where it states that the project leader possesses higher privilege, Han can overwrite the document. XrML and ODRL do not provide mechanisms for concurrent access control, as far as we know. Additionally, XrML and ODRL do not provide status constraint to describe the real-time condition of the content.

### 4.3.2 Licenses Evolution Modelling

Jack is designing a new business which involves creating offers and licenses. He is interested in knowing how the distribution of the licenses between numerous entities would influence the licenses evolution. The licenses can be recycled (e.g. *renew* right), can be transferred (e.g. *give* right) etc.

He would like to check if users are allowed to assert desirable rights, e.g. a *print* right, at their systems, and he wants to know if conflicts, i.e. if the the asserted right conflicted with the existing right stated in the license, could arise that might undermine the business. For instance, a user asserts a new *view* right which allows her to view the ebook indefinitely.

The license clause `canassert` of this offer determines if the user has the privilege to add right (i.e. license clause, the argument *C*) to this license.

```

offer(ebook:a_fiction,
[ (canview(B1, B2, User) :-
    get_value(B1, user, U), U=User,
    get_value(B1, viewed_times, PT),
    get_value(B1, max_views, Max), PT<Max,
    get_value(B1, expires, Exp),
    today(D), Exp>D),
(cancopy(B1, B2, User) :-

```

```

    get_value(B1,user,U), U=User,
    get_value(B1,expires,Exp),
    today(D), Exp>D),
(cangive(B1,B2,User1,User2) :-
  get_value(B1,user,U), U=User1,
  set_value(B1,user,User2,B2),
  set_value(B1,played_times,0,B2)),
(canagree(B1,B2,User) :-
  today(D), Exp is D+366,
  set_value(B1,expires,Exp,B2),
  set_value(B1,user,User,B2)),
(canassert(C1,C2,B1,B2,C,B,User) :-
  get_value(B1,user,U), U=User,
  append(C1,C,C2),
  append(B1,B,B2))
),
[ (viewed_times=0), (max_views=10),
  (expires=20/12/2005), (user=anyone)
]
)

```

Jack can use his license interpreter (simulator) to analyze the license evolutions. The license interpreter records each state of the license evolutions (i.e. each new license generated as a result of the evolution caused by the rules), including the newly generated licenses and the original licenses.

Additionally, through multiset rewriting the license interpreter is able to simulate the communications of more than two entities, for instance, content providers and content users. The license interpreter is able to help Jack tracing the logical design errors in his rights management system.

Jack constructs the corresponding rules for the analysis on the evolutions of this license:

```

agree(Ebook,User) :
  offer(Ebook,C,B1) ->
  offer(Ebook,C,B1), license(Ebook,C,B2)
  <= C |- canagree(B1,B2,User)
view(Ebook,User) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- canview(B1,B2,User)
copy(Ebook,User) :
  license(Ebook,C,B1) ->
  license(Ebook,C,B1), license(Ebook,C,B2)
  <= C |- cancopy(B1,B2,User)
give(Ebook,User1,User2) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- cangive(B1,B2,User1,User2)
assert(Ebook,C,B,U) :
  license(Ebook,C1,B1) -> license(Ebook,C2,B2)
  <= C1 |- canassert(C1,C2,B1,B2,C,B,U)

```

The evolution of this offer may take the form of Figure 4. An example of the license (l1) generated (at state 5 for user a who asserts a *print* right on the license) may look as follows:

```

license(ebook:a_fiction,
[ (canview(B1,B2,User) :-
  get_value(B1,user,U), U=User,
  get_value(B1,viewed_times,PT),
  get_value(B1,max_views,Max), PT<Max,
  get_value(B1,expires,Exp),
  today(D), Exp>D),
(cancopy(B1,B2,User) :-
  get_value(B1,user,U), U=User,
  get_value(B1,expires,Exp),

```

```

  today(D), Exp>D),
(cangive(B1,B2,User1,User2) :-
  get_value(B1,user,U), U=User1,
  set_value(B1,user,User2,B2),
  set_value(B1,played_times,0,B2)),
(canagree(B1,B2,User) :-
  today(D), Exp is D+366,
  set_value(B1,expires,Exp,B2),
  set_value(B1,user,User,B2)),
(canassert(C1,C2,B1,B2,C,B,User) :-
  get_value(B1,user,U), U=User,
  append(C1,C,C2),
  append(B1,B,B2)),
(canprint(B1,B2,User) :-
  get_value(B1,user,U), U=User)
],
[ (viewed_times=0), (max_views=10),
  (expires=), (user=a)
]
)

```

The license clause `canprint` is an example of the right asserted by a user, indicating that the user is allowed to print the ebook. This is done by the user (say Bob) executing the rule:

```

assert(ebook:a_fiction,
[ (canprint(B1,B2,User) :-
  get_value(B1,user,U), U=User)],
[[], bob]).

```

The license interpreter is able to generate all possible licenses and offers. As can be seen at state 2, the *copy* right is performed on the old license (l1) to generate a new license (l2) in addition to the original license (l1) at state 3.

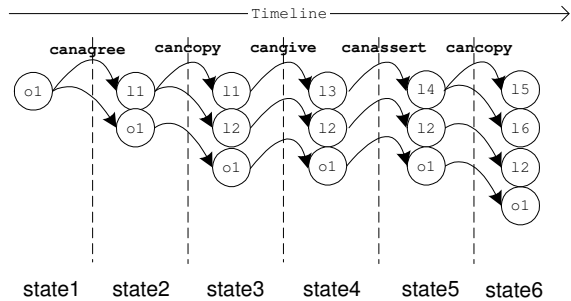


Figure 4: A state chart of an example of license evolution in this scenario.

The license interpreter can in principle be used as a licensing model checker, and the LicenseScript language can be used as model language to specify the licensing processes. In short, LicenseScript is a potential modelling language for licensing processes.

Additionally, LicenseScript allows dynamic generation of new vocabulary for the rights expression in the license clause, as shown in this scenario. XrML and ODRL do not support this feature. However, this has to be implemented with great care because this could be abusively exploited, which undermines the rights management system. We may control the rules with care, to render the rules trusted.

## 4.4 Final Remarks

Table 4 summarizes the essence of all the scenarios presented in the XML-based RELs specifications (including the scenarios not discussed in this paper).

The rows of the table correspond to the anatomy presented in section 2. The columns correspond to the scenarios, which are: (1) ODRL scenarios: ebook #1 (E1), #2 (E2), #3 (E3), video (VD), superdistribution (SD), software(SW), image(IM) and audio(AD); (2) XrML scenarios: preview/promotional (PP), subscription (SB), territory restriction (TR), temporal ordering of rights (TO), usage of part of a work (PW), site license (SL), personal lending (PL) and giving (GV), superdistribution (SD), unrestricted sales (US), personal copies (PC), web service access (WS), software execution (SW), confidentiality of rights (CR), operational model (OM) and secure device (SV); and (3) Novel scenarios: project documents sharing (PS) and license evolution modelling (LM).

We put a ‘✓’ where we can show that LicenseScript covers the corresponding feature of the REL. This shows that LicenseScript is more flexible and expressive than XML-based RELs.

## 5 Conclusions and Future Work

In this paper we have presented a simple way to analyse the anatomy of a Right Expression Language. In addition, we have studied the scenarios presented in XrML and ODRL, and translated them into LicenseScript (the REL we proposed in [3]). We have also studied some novel scenarios and formalized them in LicenseScript.

We think that this investigation is useful for understanding the strengths and weaknesses of ODRL, XrML and LicenseScript, and for assessing their capability of describing a number of important content access and distribution patterns as well as licensing processes.

We have also demonstrated that LicenseScript is sufficiently flexible and expressive to express the scenarios (studied so far) elegantly and effortlessly.

We also highlighted the fact that in LicenseScript one can define a new set of vocabulary for rights expression in the license clause, which XML-based RELs cannot support. We believe that this feature of LicenseScript may support copyrights enforcement in the rights management system. This deserves further study as our future work.

## Acknowledgement

We like to thank Ernst-Jan Goedvolk from Telematica Instituut and Mark Stefik from Xerox PARC for their

valuable help.

## References

- [1] J-P. Banâtre, P. Fradet, and D. L. Métayer. Gamma and the chemical reaction model: Fifteen years after. In C. Calude, G. Paun, G. Rozenberg, and A. Salomaa, editors, *Workshop on Multiset Processing (WMP)*, volume 2235 of *Lecture Notes in Computer Science*, pages 17–44. Springer-Verlag, Berlin, August 2001.
- [2] L. Jean Camp. DRM: doesn’t really mean digital copyright management. In *Proceedings of the 9th ACM conference on Computer and Communications Security*, pages 78–87. ACM Press, 2002.
- [3] 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 *3rd International Conference on Web Delivering of Music (WEDELMUSIC)*, page to appear, Los Alamitos, California, United States, September 2003. IEEE Computer Society Press.
- [4] C. Gunter, S. Weeks, and A. Wright. Models and languages for digital rights. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, pages 4034–4038, Maui, Hawaii, United States, January 2001. IEEE Computer Society Press.
- [5] H. Guo. Digital rights management (DRM) using XrML. In *T-110.501 Seminar on Network Security 2001*, page Poster paper 4, 2001. <http://www.tml.hut.fi/Studies/T-110.501/2001/papers/>.
- [6] R. Iannella. Open digital rights management. In *World Wide Web Consortium (W3C) DRM Workshop*, page Position paper 23, January 2001. <http://www.w3.org/2000/12/drm-ws/pp/>.
- [7] D. Mulligan and A. Burstein. Implementing copyright limitations in rights expression languages. In J. Feigenbaum, editor, *Proceedings of 2002 ACM CCS-9 Workshop on Security and Privacy in Digital Rights Management*, volume 2696 of *Lecture Notes in Computer Science*, page To appear. Springer-Verlag, November 2002.
- [8] David Parrott. Requirements for a rights data dictionary and rights expression language. Technical Report version 1.0, Reuters Ltd., 85 Fleet St., London EC4P 4AJ, June 2001. In response to ISO/IEC JTC1/SC29/WG11 N4044: “Reissue

Table 4: The symbol '✓' indicates the scenario exhibits the corresponding feature.

Anatomy		ODRL Scenarios									XrML Scenarios														Novel Scenarios		
Categories	LS	E1	E2	E3	VD	SD	SW	IM	AD	PP	SB	TR	TO	PW	SL	PL	GV	SD	US	PC	WS	SW	CR	OM	SV	PS	LM
Right-Render	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Right-Reuse	✓			✓			✓							✓				✓								✓	
Right-Transport	✓			✓		✓			✓							✓	✓	✓		✓						✓	✓
Right-Manage Object	✓																			✓	✓					✓	✓
Right-Regulate Rights	✓																									✓	✓
Obligation	✓	✓	✓		✓		✓			✓	✓	✓		✓			✓	✓					✓		✓		
Object-Generalized Types	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Object-Classification	✓											✓		✓													
Object-Delivery Methods	✓	✓	✓		✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓							
Object-Fuzzy Match																											
Subject	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Constraint-Temporal	✓			✓		✓	✓	✓				✓	✓	✓	✓	✓	✓		✓	✓					✓	✓	✓
Constraint-Bound	✓	✓	✓	✓	✓		✓	✓												✓				✓			
Constraint-Environment	✓		✓				✓						✓									✓			✓		
Constraint-Aspect	✓			✓	✓																			✓			
Constraint-Purpose	✓								✓																		
Constraint-Status	✓																									✓	
Relation-Ordering-Ante Obs	✓	✓	✓		✓		✓			✓	✓			✓			✓	✓								✓	✓
Relation-Ordering-Cons Obs	✓																										
Relation-Ordering-Total	✓											✓	✓														
Relation-Ordering-Partial	✓											✓										✓					
Relation-Association	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Relation-Naming	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Relation-Limitation	✓			✓		✓	✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓
Relation-Characteristic	✓			✓	✓																			✓			
Model-Revenue-Pay peruse	✓				✓					✓	✓							✓									
Model-Revenue-Pay upfront	✓	✓	✓				✓																				
Model-Revenue-Pay flatrate	✓												✓	✓			✓	✓									
Model-Revenue-Tiered Pay	✓									✓																	
Model-Revenue-Pay Multi	✓	✓			✓																						
Model-Revenue-Fraction Pay	✓	✓			✓																						
Model-Provision-Conflicts	✓																									✓	✓
Model-Provision-Alternatives																											
Model-Provision-Defaults																											
Model-Operational	✓																								✓		
Model-Contract	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Model-Copyrights																											
Model-Security-Identification	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Model-Security-Authentication	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓						✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Model-Security-Authorization	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓						✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Model-Security-Access Control	✓																										✓
Model-Security-Confidentiality	✓																							✓			
Model-Security-Nonrepudiation	✓									✓	✓		✓	✓	✓				✓	✓	✓			✓	✓		
Model-Security-Integrity	✓											✓								✓							
Model-Security-Audit Trails	✓													✓		✓	✓	✓	✓							✓	
Model-Security-Privacy	✓																						✓				

of the Call for Requirements for a Rights Data Dictionary and a Rights Expression Language” – MPEG-21.

[9] R. Pucella and V. Weissman. A logic for reasoning about digital rights. In *IEEE Proceedings of the Computer Security Foundations Workshop*, pages 282–294, Cape Breton, Nova Scotia, Canada, June 2002. IEEE Computer Society Press.

[10] B. Rosenblatt, B. Trippe, and S. Mooney. *Digital Rights Management: Business and Technology*. John Wiley & Sons, New York, United States, November 2002. ISBN 0764548891.

[11] P. Samuelson. Digital rights management {and,or,vs.} the law. *Communications of ACM*, 46(4):41–45, April 2003.

## A Appendix: ODRL Scenarios

ODRL code for ebook scenarios #1, #2 and #3, video scenario, super-distribution scenario, software scenario, image scenario, and audio scenario. LicenseScript code for video scenario, super-distribution scenario, software scenario, image scenario, and audio scenario.

### A.1 ODRL Ebook Scenario #1

ODRL code:

```
<?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
xmlns:marc=http://www.loc.gov/marc/>
<o-ex:offer>
<o-ex:asset><o-ex:context><o-dd:uid>
urn:ebook.world/999999/ebook/rossi-000001
</o-dd:uid>
<o-dd:name>
Why Cats Sleep and We Dont
</o-dd:name>
</o-ex:context></o-ex:asset>
<o-ex:permission>
<o-dd:display>
<o-ex:constraint>
<o-dd:cpu/>
</o-ex:constraint>
</o-dd:display>
<o-dd:print>
<o-ex:constraint>
<o-dd:count>2</o-dd:count>
</o-ex:constraint>
</o-dd:print>
<o-ex:requirement>
<o-dd:prepay><o-dd:payment>
<o-dd:amount o-dd:currency=AUD>20.00</o-dd:amount>
<o-dd:taxpercent o-dd:code=GST>10.00</o-dd:taxpercent>
</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>1</o-dd:min>
<o-dd:max>5</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>
x500:c=AU;o=RightsDir;cn=CorkyRossi
</o-dd:uid>
<o-dd:role>onix:roles/A01</o-dd:role>
</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>60</o-dd:percentage>
</o-ex:rightsholder>
</o-ex:party>
<o-ex:party>
<o-ex:context>
<o-dd:uid>
x500:c=AU;o=RightsDir;cn=AddisonRossi
</o-dd:uid>
<o-dd:role>onix:roles/A12</o-dd:role>
</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>10</o-dd:percentage>
</o-ex:rightsholder>
</o-ex:party>
<o-ex:party>
<o-ex:context>
<o-dd:uid>
x500:c=AU;o=RightsDir;cn=EBooksRUS
</o-dd:uid>
<o-dd:role>marc:roles/pbl</o-dd:role>
</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>30</o-dd:percentage>
</o-ex:rightsholder>
</o-ex:party>
</o-ex:offer>
```

LicenseScript code and rules are provided in section 4.1.1.

### A.2 ODRL Ebook Scenario #2

ODRL code:

```
<?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>
urn:ebook.world/999999/license/1234567890-ABCDEF
</o-dd:uid>
<o-dd:pLocation>
Sydney, Australia
</o-dd:pLocation>
<o-dd:remark>
Transacted by Example.Com
</o-dd:remark>
</o-ex:context>
<o-ex:asset>
<o-ex:context><o-dd:uid>
urn:ebook.world/999999/ebook/rossi-000001
</o-dd:uid></o-ex:context>
</o-ex:asset>
<o-ex:permission>
<o-dd:display>
<o-ex:constraint><o-dd:cpu>
<o-ex:context><o-dd:uid>
Adobe-WebBuy:CPD-ID:ER-393939-DSS-787878
</o-dd:uid></o-ex:context>
</o-dd:cpu></o-ex:constraint>
```

```

</o-dd:display>
<o-dd:print>
<o-ex:constraint>
<o-dd:count>2</o-dd:count>
</o-ex:constraint>
</o-dd:print>
<o-ex:requirement>
<o-dd:prepay><o-dd:payment>
<o-dd:amount o-dd:currency=AUD>
20.00
</o-dd:amount>
<o-dd:taxpercent o-dd:code=GST>
10.00
</o-dd:taxpercent>
</o-dd:payment></o-dd:prepay>
</o-ex:requirement>
</o-ex:permission>
<o-ex:party><o-ex:context>
<o-dd:uid>
urn:ebook.world/999999/users/msmth-000111
</o-dd:uid>
<o-dd:name>Mary Smith</o-dd:name>
</o-ex:context></o-ex:party>
</o-ex:agreement>
</o-ex:rights>

```

LicenseScript code and rules are provided in section 4.1.2.

### A.3 ODRL Ebook Scenario #3

ODRL code:

```

<?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
xmlns:ebx=http://www.ebxwg.org/ebook/vocab/
xmlns:marc=http://www.loc.gov/marc/>
<o-ex:context>
<o-dd:uid>
urn:ebook.world/999999/voucher/2001/1234567890
</o-dd:uid>
<o-dd:date>
<o-dd:fixed>2001-05-01T08:30:00</o-dd:fixed>
</o-dd:date>
<o-dd:event>issued</o-dd:event>
</o-ex:context>
<o-ex:asset o-ex:id=a001>
<o-ex:context>
<o-dd:uid>isbn:872-2345-981</o-dd:uid>
<o-dd:name>XML: A Managers Guide</o-dd:name>
</o-ex:context>
</o-ex:asset>
<o-ex:party>
<o-ex:context>
<o-dd:uid>
http://publishers.net/registry/AWL
</o-dd:uid>
<o-dd:name>Addison-Wesley </o-dd:name>
<o-dd:reference>
http://www.addison-wesley.com
</o-dd:reference>
</o-ex:context>
<o-ex:rightsholder/>
</o-ex:party>
<o-ex:agreement>
<o-ex:asset o-ex:idref=a001/>
<o-ex:party>
<o-ex:context>
<o-dd:uid>
http://distributors.net/registry/xyz
</o-dd:uid>
<o-dd:name>XYZ Company </o-dd:name>
<o-dd:role>marc:dst</o-dd:role>
</o-ex:context>
</o-ex:party>

```

```

<o-ex:permission>
<o-dd:sell>
<o-ex:constraint>
<o-dd:count>5000</o-dd:count>
</o-ex:constraint>
</o-dd:sell>
</o-ex:permission>
</o-ex:agreement>
<o-ex:agreement>
<o-ex:asset o-ex:idref=a001/>
<o-ex:party>
<o-ex:context>
<o-dd:uid>
http://people.net/registry/john-doe-9999
</o-dd:uid>
<o-dd:name>John Doe</o-dd:name>
</o-ex:context>
</o-ex:party>
<o-ex:permission>
<o-ex:constraint>
<o-dd:datetime>
<o-dd:start>2001-01-01T00:00:00</o-dd:start>
<o-dd:end>2004-12-31T23:59:59</o-dd:end>
</o-dd:datetime>
</o-ex:constraint>
<o-dd:display>
<o-ex:constraint><o-dd:accumulated>
P30D
</o-dd:accumulated></o-ex:constraint>
</o-dd:display>
<o-dd:print>
<o-ex:container o-ex:type=and>
<o-ex:constraint>
<o-dd:count>5</o-dd:count>
<o-dd:printer>
<o-ex:context>
<o-dd:uid>
guid:TrustPrint/4747474742222
</o-dd:uid>
</o-ex:context>
</o-dd:printer>
</o-ex:constraint>
<o-ex:constraint>
<o-dd:unit o-ex:type=onix:NumberOfPages>
<o-ex:constraint>
<o-dd:count>100</o-dd:count>
</o-ex:constraint>
</o-dd:unit>
<o-dd:printer/>
</o-ex:constraint>
<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>100</o-dd:max>
</o-dd:range>
<o-dd:count>5</o-dd:count>
</o-ex:constraint>
</o-dd:unit>
<o-dd:interval>P7D</o-dd:interval>
<o-dd:printer/>
</o-ex:constraint>
</o-ex:container>
</o-dd:print>
<o-dd:excerpt>
<o-ex:constraint>
<o-dd:memory>
<o-ex:container o-ex:type=and>
<o-ex:constraint>
<o-dd:unit o-ex:type=ebx:NumberOfBytes>
<o-ex:constraint>
<o-dd:count>
1000000
</o-dd:count>
</o-ex:constraint>
</o-dd:unit>
</o-ex:constraint>
<o-ex:constraint>
<o-dd:unit o-ex:type=ebx:NumberOfBytes>
<o-ex:constraint>

```



```

<o-dd:count>
5000
</o-dd:count>
<o-dd:interval>
P7D
</o-dd:interval>
</o-ex:constraint>
</o-dd:unit>
</o-ex:constraint>
</o-ex:container>
</o-dd:memory>
</o-ex:constraint>
</o-dd:excerpt>
<o-dd:give>
<o-ex:constraint>
<o-dd:datetime>
<o-dd:start>
2002-01-01T00:00:00
</o-dd:start>
</o-dd:datetime>
</o-ex:constraint>
</o-dd:give>
</o-ex:permission>
</o-ex:agreement>
</o-ex:rights>

```

LicenseScript code and rules are provided in section 4.1.3.

## A.4 Video Scenario

ODRL code:

```

<?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:mpeg7=http://www.mpeg7.org/2001/MPEG-7_Schema>
<o-ex:context>
<o-dd:uid>doi:/voucher/383838383</o-dd:uid>
<o-dd:name>
The Voucher for XML: The Movie</o-dd:name>
<o-dd:dLocation>
http://example.com/odrl/383838383.xml
</o-dd:dLocation>
</o-ex:context>
<o-ex:offer>
<o-ex:asset><o-ex:context>
<o-dd:uid>doi:0.9999999/video/383838383</o-dd:uid>
<o-dd:name>XML: The Movie</o-dd:name>
</o-ex:context></o-ex:asset>
<o-ex:party><o-ex:context>
<o-dd:uid>
x500:c=IT;o=Registry;cn=MassimoCanale
</o-dd:uid>
</o-ex:context><o-ex:rightsholder>
<o-dd:percentage>75</o-dd:percentage>
</o-ex:rightsholder>
</o-ex:party>
<o-ex:party>
<o-ex:context>
<o-dd:uid>
x500:c=IT;o=Registry;cn=SimonaCanale
</o-dd:uid>
</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>25</o-dd:percentage>
</o-ex:rightsholder>
</o-ex:party><o-ex:context>
<o-dd:uid>x500:c=IT;o=Registry;cn=MariaCanale
</o-dd:uid>
</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>10</o-dd:percentage>
</o-ex:rightsholder>
</o-ex:party>
</o-ex:party>
</o-ex:permission>

```

```

<o-dd:play><o-ex:constraint>
<o-dd:quality o-ex:type=mpeg7:resolution>
<o-ex:constraint>
<o-dd:range>
<o-dd:max>30</o-dd:max>
</o-dd:range>
</o-ex:constraint>
</o-dd:quality>
</o-ex:constraint>
<o-ex:requirement>
<o-dd:peruse>
<o-dd:payment>
<o-dd:amount o-dd:currency=ITL>
1000.00
</o-dd:amount>
</o-dd:payment>
</o-dd:peruse>
</o-ex:requirement>
</o-dd:play>
<o-dd:play><o-ex:constraint>
<o-dd:quality o-ex:type=mpeg7:resolution>
<o-ex:constraint>
<o-dd:range>
<o-dd:max>90.0</o-dd:max>
</o-dd:range>
</o-ex:constraint>
</o-dd:quality>
</o-ex:constraint>
<o-ex:requirement>
<o-dd:peruse>
<o-dd:payment>
<o-dd:amount o-dd:currency=ITL>
5000.00</o-dd:amount>
</o-dd:payment>
</o-dd:peruse>
</o-ex:requirement>
</o-dd:play>
</o-ex:permission>
</o-ex:offer>
</o-ex:rights>

```

LicenseScript code:

```

license(doi:0.9999999/video/383838383,
[ (canplay (B1,B2,R) :-
  R<=30,
  get_value (B1,paid_high,Paid), paid=true),
  (canplay (B1,B2,R) :-
  R>30, R<=90,
  get_value (B1,paid_low,Paid), paid=true),
  (canpay (B1,B2,H1,H2,H3,R) :-
  get_value (B1,rate_low,Rate),
  get_value (B1,rightsholder1,Holder1),
  get_value (B1,rightsholder2,Holder2),
  get_value (B1,rightsholder3,Holder3),
  set_value (B1,paid_low,true,B2)),
  (canpay (B1,B2,H1,H2,H3,R) :-
  get_value (B1,rate_high,Rate),
  get_value (B1,rightsholder1,Holder1),
  get_value (B1,rightsholder2,Holder2),
  get_value (B1,rightsholder3,Holder3),
  set_value (B1,paid_high,true,B2)) ]),
[ (currency=ITL), (rate_low=1000.00),
(rate_high=5000.00),
(rightsholder1="x500:c=IT;o=Registry;cn=MassimoCanale"),
(rightsholder2="x500:c=IT;o=Registry;cn=SimoneaCanale"),
(rightsholder3="x500:c=IT;o=Registry;cn=MariaCanale"),
(paid_high=false), (paid_low=false) ])

wallet (
[ (canload (Bw1,Bw2,A) :-
  get_value (Bw1,money,Money),
  N is A+Money,
  set_value (Bw1,money,N,Bw2)),
  (cantransfer (Bw1,Bw2,H1,H2,H3,R) :-
  get_value (Bw1,money,Money),
  Money >= R,
  transfer (Holder1,Rate*0.75),
  transfer (Holder2,Rate*0.15),
  transfer (Holder3,Rate*0.10)),

```

```

    N is Money-R,
    set_value(Bw1,money,N,Bw2) ],
[ (money=x) ]

```

### LicenseScript rules:

```

play(Video,Res) :
  license(Video,C,B1) -> license(Video,C,B2)
<= C |- canplay(Video,B1,B2,Res)
pay(Video) :
  license(Video,C,B1), wallet(Cw,Bw1) ->
  license(Video,C,B2), wallet(Cw,Bw2)
<= C |- canpay(B1,B2,H1,H2,H3,R),
  Cw |- cantransfer(Bw1,Bw2,H1,H2,H3,R)

```

## A.5 Superdistribution Scenario

```

<?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:asset><o-ex:context>
<o-dd:uid>doi:10.9999999/voucher/383838383
</o-dd:uid>
<o-dd:name>The Voucher for XML: The Movie
</o-dd:name>
</o-ex:context>
<o-ex:asset>
<o-ex:party><o-ex:context>
<o-dd:uid>x500:c=US;o=Example;cn=JJJones</o-dd:uid>
</o-ex:context>
</o-ex:party>
<o-ex:permission><o-dd:give>
<o-ex:constraint>
<o-dd:datetime>
<o-dd:end>2001-12-31T23:59:59</o-dd:end>
</o-dd:datetime>
</o-ex:constraint>
</o-dd:give>
</o-ex:permission>
</o-ex:agreement>
</o-ex:rights>

```

### LicenseScript code:

```

license(doi:10.9999999/voucher/383838383,
[ (cangive(B1,B2,D) :-
  get_value(B1,expires,Expdate),
  today(Today), Expdate>Today,
  get_value(B1,in_domain,Id1),
  identify(Id2), Id1=Id2,
  get_value(B1,to_domain,Id3), Id3=any,
  set_value(B1,in_domain,D,B2),
  set_value(B1,to_domain,Id1,B2)
],
[ (in_domain=x500:c=US;o=Example;cn=JJJones),
(to_domain=any),
(expires=2001-12-31T23:59:59) ])

```

### LicenseScript rule:

```

give(Video,Domain) :
  license(Video,C,B1) -> license(Video,C,B2)
<= C |- cangive(B1,B2,Domain)

```

## A.6 Software Scenario

### ODRL code:

```

<?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:offer><o-ex:asset>

```

```

<o-ex:context>
<o-dd:uid>
urn:example.com/learnobject/uni/aa/7373722
</o-dd:uid>
<o-dd:name>XML: The Lecture</o-dd:name>
</o-ex:context>
</o-ex:asset>
<o-ex:party><o-ex:context>
<o-dd:uid>
x500:c=AU;o=UniA;cn=ProfBean
</o-dd:uid>
</o-ex:context>
<o-ex:rightsholder>
<o-ex:container o-ex:type=and>
<o-dd:percentage>50</o-dd:percentage>
<o-ex:permission o-ex:idref=p001/>
</o-ex:container>
<o-ex:container o-ex:type=and>
<o-dd:percentage>25</o-dd:percentage>
<o-ex:permission o-ex:idref=p002/>
</o-ex:container>
</o-ex:rightsholder>
</o-ex:party>
<o-ex:permission o-ex:id=p001>
<o-dd:execute>
<o-ex:constraint>
<o-dd:network/>
</o-ex:constraint>
<o-ex:requirement>
<o-dd:peruse><o-dd:payment>
<o-dd:amount o-dd:currency=AUD>
2.00</o-dd:amount>
</o-dd:payment>
</o-dd:peruse>
</o-ex:requirement>
</o-dd:execute>
</o-ex:permission>
<o-ex:permission o-ex:id=p002>
<o-dd:lend>
<o-ex:constraint>
<o-dd:individual><o-ex:constraint>
<o-dd:count>100</o-dd:count>
</o-ex:constraint></o-dd:individual>
</o-ex:constraint>
<o-ex:requirement>
<o-dd:prepay><o-dd:payment>
<o-dd:amount o-dd:currency=AUD>
1000.00
</o-dd:amount>
</o-dd:payment></o-dd:prepay>
</o-ex:requirement>
</o-dd:lend>
</o-ex:permission>
</o-ex:offer>

```

### LicenseScript code:

```

license(urn:example.com/learnobject/uni/aa/7373722,
[ (canexecute(B1,B2) :-
  get_value(B1,paid,Paid), Paid=true,
  set_value(B1,paid,false,B2)),
(canlend(B1,B2,D) :-
  get_value(B1,lend_counts,Counts),
  Counts<=100, get_value(B1,in_domain,Id1),
  identify(Id2), Id1=Id2,
  set_value(B1,in_domain,D,B2),
  set_value(B1,to_domain,Id1,B2),
  Newcnt is Counts+1,
  set_value(B1,lend_counts,Newcnt,B2)) ],
(canpay(B1,B2,H,R) :-
  get_value(B1,paid,Paid), Paid=false,
  get_value(B1,rightsholder,H),
  get_value(B1,execute_rate,R),
  set_value(B1,paid,true,B2)),
[ (currency=AUD), (lend_rate=1000.00),
(execute_rate=2.00), (lend_counts=0),
(rightsholder="x500:c=AU;o=UniA;cn=ProfBean"),
(paid=false), (to_domain=any),
(in_domain="x500:c=AU;o=UniA;cn=ProfBean") ])

```

```
wallet (
[ (canload(Bw1,Bw2,A) :-
  get_value(Bw1,money,Money),
  N is A+Money,
  set_value(Bw1,money,N,Bw2)),
(canttransfer(Bw1,Bw2,H,R) :-
  get_value(Bw1,money,Money),
  Money >= R, transfer(H,R),
  N is Money-R,
  set_value(Bw1,money,N,Bw2)) ],
[ (money=x) ])
```

### LicenseScript rules:

```
execute(Sware) :
  license(Sware,C,B1) -> license(Sware,C,B2)
<= C |- canexecute(B1,B2)
lend(Sware,Domain) :
  license(Sware,C,B1) -> license(Sware,C,B2)
<= C |- canlend(B1,B2,Domain)
pay(Sware,H,R) :
  license(Sware,C,B1), wallet(Cw,Bw1) ->
  license(Sware,C,B2), wallet(Cw,Bw2)
<= C |- canpay(B1,B2,H,R),
  Cw |- canttransfer(Bw1,Bw2,H,R)
```

## A.7 Image Scenario

### ODRL code:

```
<?xml version="1.0" encoding=UTF-8?>
<rdf:RDF
  xmlns:prism="http://prismstandard.org/namespaces/basic/4.0/"
  xmlns:prl="http://prismstandard.org/namespaces/prl/1.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
  <rdf:Description rdf:about=
    "http://wanderlust.com/2000/08/Corfu.jpg">
    <dc:title>
      Walking on the Beach in Corfu
    </dc:title>
    <dc:creator>John Peterson</dc:creator>
    <dc:contributor>Sally Smith, lighting</dc:contributor>
    <dc:format>image/jpeg</dc:format>
    <dc:identifier
      rdf:resource="http://wanderlust.com/content/2357845"/>
    <o-ex:rights><o-ex:agreement>
    <o-ex:permission>
    <o-dd:display><o-ex:constraint>
    <o-dd:industry o-ex:type="prism:vocabs/SIC/6532"/>
    <o-dd:spatial o-ex:type="prism:vocabs/ISO-3166/US">
    <o-ex:constraint>
    <o-dd:datetime>
    <o-dd:start>
      2001-01-01T00:00:00
    </o-dd:start>
    </o-dd:datetime>
    </o-ex:constraint></o-dd:spatial>
    <o-dd:spatial o-ex:type=prism:vocabs/ISO-3166/GR>
    <o-ex:constraint>
    <o-dd:datetime>
    <o-dd:end>2003-12-31T23:59:59</o-dd:end>
    </o-dd:datetime>
    </o-ex:constraint>
    </o-dd:spatial></o-ex:constraint>
    </o-dd:display>
    </o-ex:permission>
    <o-ex:party><o-ex:context>
    <o-dd:uid>
      urn:prism.orgs:cool-cat-magazine
    </o-dd:uid>
    </o-ex:context>
    </o-ex:party>
    </o-ex:agreement>
    </o-ex:rights>
  </rdf:Description>
</rdf:RDF>
```

### LicenseScript code:

```
lic(http://wanderlust.com/2000/08/Corfu.jpg,
[ (canpublish(B1,B2,D) :-
  today(Today), get_value(B1,starts,Starts),
  get_value(B1,ends,Ends),
  Starts<=Today, Ends>Today,
  get_value(B1,in_domain,ValidD), ValidD=D) ],
[ (in_domain="urn:prism.orgs:cool-cat-magazine"),
  (starts="2001-01-01T00:00:00"),
  (ends="2003-12-31T23:59:59") ])
```

### LicenseScript rule:

```
publish(Img,Domain) :
  license(Img,C,B1) -> license(Img,C,B2)
<= C |- canpublish(B1,B2,Domain)
```

## A.8 Audio Scenario

### ODRL code:

```
<?xml version="1.0" encoding="UTF-8"?>
<di:DIDL
  xmlns:di="urn:mpeg:mpeg21:2002/01-DIDL-NS"
  xmlns:diid="urn:mpeg:mpeg21:2002/01-DIID-NS"
  xmlns:sector="urn:sectors:2002:vocab"
  xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
  <di:Item>
    <di:Descriptor>
      <di:Statement type="text/text">
        Cool Cats CD Single
      </di:Statement>
    </di:Descriptor>
    <di:Descriptor>
      <di:Statement type="text/xml">
        <diid:identifier>
          A1-888999-0029733-22-F
        </diid:identifier>
      </di:Statement>
    </di:Descriptor>
    <di:Descriptor>
      <di:Statement type="text/xml">
        <o-ex:rights><o-ex:offer>
          <o-ex:permission><o-dd:sell>
            <o-dd:transferPerm o-dd:downstream=equal>
            <o-dd:play>
            <o-ex:constraint>
            <o-dd:purpose o-ex:type=sectors:educational/>
          </o-ex:constraint>
        </o-dd:play>
        </o-dd:transferPerm>
      </o-dd:sell></o-ex:permission>
    </o-ex:offer>
    </o-ex:rights>
      </di:Statement>
    </di:Descriptor>
  </di:Item>
</di:DIDL>
```

### LicenseScript code:

```
lic(A1-888999-0029733-22-F,
[ (cansell(B1,B2,D,Dt) :-
  get_value(B1,domain_type,Domtype),
  Dt=Domtype,
  set_value(B1,in_domain,D,B2)),
  (canplay(B1,B2) :-
  identify(Id1),
  get_value(B1,in_domain,Id2),
  Id1=Id2) ],
[ (domain_type="sectors:educational"),
  (to_domain=any), (in_domain=any) ])
```

### LicenseScript rules:

```

sell(Audio,Domain,Type) :
  license(Audio,C,B1) -> license(Audio,C,B2)
  <= C |- cansell(B1,B2,Domain,Type)
play(Audio) :
  license(Audio,C,B1) -> license(Audio,C,B2)
  <= C |- canplay(B1,B2)

```

## B Appendix: XrML Scenarios

XrML code for unlimited/limited usage, preview/promotional model, tiered pricing model, pay per view, subscription, territory restriction, temporal ordering of rights, usage of part of a work, payment according to different users, site license, pay multi-entities, personal lending and giving, superdistribution, unrestricted sales, personal copies, fraction payments, web service access, software execution, confidentiality of rights, operational model and secure device. LicenseScript code for unlimited/limited usage, preview/promotional model, tiered pricing model, pay per view, subscription, territory restriction, temporal ordering of rights, usage of part of a work, payment according to different users, site license, pay multi-entities, personal giving, unrestricted sales, personal copies, fraction payments, web service access, software execution, confidentiality of rights, operational model and secure device.

### B.1 Unlimited Usage

XrML code:

```

-<license>
-<inventory>
-<keyHolder licensePartId="issuedToParty">
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQQA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
-<cx:digitalWork licensePartId="eBook">
-<cx:locator>
<nonSecureIndirect
URI="http://www.contentguard.com/sampleBook.spd"/>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grant>
-<keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
<obtain/>
-<grantGroup>
-<keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
-<grant>
<cx:play/>
<cx:digitalWork licensePartIdRef="eBook"/>
</grant>
-<grant>
<cx:print/>
<cx:digitalWork licensePartIdRef="eBook"/>
</grant>
-<grant>

```

```

<cx:copy/>
<cx:digitalWork licensePartIdRef="eBook"/>
</grant>
-<grant>
<cx:extract/>
<cx:digitalWork licensePartIdRef="eBook"/>
</grant>
</grantGroup>
-<sx:fee>
-<sx:paymentPerUse>
<sx:rate currency="USD">25.99</sx:rate>
</sx:paymentPerUse>
</sx:fee>
</grant>
</license>

```

LicenseScript code:

```

license(http://www.contentguard.com/sampleBook.spd,
[
  (canplay(B1,B2) :-
    get_value(B1,paid,Paid), Paid=true,
    set_value(B1,paid,false,B2)),
  (canprint(B1,B2) :-
    get_value(B1,paid,Paid), Paid=true,
    set_value(B1,paid,false,B2)),
  (cancopy(B1,B2) :-
    get_value(B1,paid,Paid), Paid=true,
    set_value(B1,paid,false,B2)),
  (canextract(B1,B2) :-
    get_value(B1,paid,Paid), Paid=true,
    set_value(B1,paid,false,B2)),
],
[
  (paid=false),
  (holder="<mod>...<exp>...")
])

```

LicenseScript rules:

```

play(Ebook) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- canplay(B1,B2)
print(Ebook) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
  <= C |- canprint(B1,B2)
copy(Ebook) :
  license(Ebook,C,B1) ->
  license(Ebook,C,B1), license(Ebook,C,B2)
  <= C |- cancopy(B1,B2)
extract(Ebook) :
  license(Ebook,C,B1) ->
  license(Ebook,C,B1), license(Ebook,C,B2)
  <= C |- canextract(B1,B2)

```

### B.2 Limited Usage

XrML code:

```

-<license>
-<inventory>
+<keyHolder licensePartId="issuedToParty">
-<cx:digitalWork licensePartId="eBook">
-<cx:locator>
<nonSecureIndirect
URI="http://www.contentguard.com/sampleBook.spd"/>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grantGroup>
-<keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
-<sx:fee>
<sx:paymentFlat>
<sx:rate currency="USD">5.99</sx:rate>
<sx:paymentRecord>
<sx:stateReference>
<uddi>
<serviceKey>

```

```

<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
<-grant>
<cx:play/>
<cx:digitalWork licensePartIdRef="eBook"/>
<-sx:validityIntervalFloating>
<-sx:stateReference>
<-uddi>
<-serviceKey>
<uuid>1F8903B0-FC03-4c5b-A445-AAFCCEC01333</uuid>
</serviceKey>
</uddi>
<-serviceParameters>
<-datum>
<cx:digitalWork licensePartIdRef="eBook"/>
</datum>
<-datum>
<-keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:validityIntervalFloating>
</grant>
<-grant>
<cx:print/>
<cx:digitalWork licensePartIdRef="eBook"/>
<-sx:exerciseLimit>
<-sx:stateReference>
<-uddi>
<-serviceKey>
<uuid>E55129C1-74DF-40d0-B2A6-367AAAB6AB05</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:exerciseLimit>
</grant>
</grantGroup>
</license>

```

### LicenseScript code:

```

license (http://www.contentguard.com/sampleBook.spd,
[ (canplay (B1, B2) :-
  get_value (B1, skey2, Key),
  get_value (B1, keyholder, Holder),
  validity (Key, Holder)),
  (canprint (B1, B2) :-
  get_value (B1, skey3, Key),
  get_value (B1, keyholder, Holder),
  limit (Key, Holder)),
  (canpay (B1, B2, Holder, Key, Rate) :-
  get_value (B1, keyholder, Holder),
  get_value (B1, skey1, Key),
  get_value (B1, rate, Rate),
  set_value (B1, paid, true, B2))
],
[ (paid=false), (rate=5.99), (currency=usd),
  (keyholder="<mod>...<exp>..."),
  (skey1="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
  (skey2="1F8903B0-FC03-4c5b-A445-AAFCCEC01333"),
  (skey3="E55129C1-74DF-40d0-B2A6-367AAAB6AB05"),
])
wallet (
[ (canload (Bw1, Bw2, A) :-
  get_value (Bw1, money, Money),
  N is A+Money,
  set_value (Bw1, money, N, Bw2)),
  (cantransfer (Bw1, Bw2, H, K, R) :-
  get_value (Bw1, money, Money),
  Money >= R, uddi_transfer (H, R),
  N is Money-R,
  set_value (Bw1, money, N, Bw2)) ],
[ (money=x) ])

```

### LicenseScript rules:

```

play (Ebook) :
  license (Ebook, C, B1) -> license (Ebook, C, B2)
  <= C |- canplay (B1, B2)
print (Ebook) :
  license (Ebook, C, B1) -> license (Ebook, C, B2)
  <= C |- canprint (B1, B2)
pay (Ebook) :
  license (Ebook, C, B1), wallet (Cw, Bw1) ->
  license (Ebook, C, B2), wallet (Cw, Bw2)
  <= C |- canpay (B1, B2, H, K, R),
  Cw |- cantransfer (B1, B2, H, K, R)

```

## B.3 Preview/Promotional View

### XrML code:

```

<-license>
<-inventory>
<-cx:digitalWork licensePartId="eBook">
<cx:description
xml:lang="en">A very good book</cx:description>
<cx:description
xml:lang="fr">Un tres bon livre</cx:description>
<-cx:metadata>
<-xml>
<-cx:simpleDigitalWorkMetadata>
<cx:title>A Book of James</cx:title>
<cx:creator>James the first</cx:creator>
<cx:copyright>
Copyright 1999 Harper Collins Publishers
</cx:copyright>
</cx:simpleDigitalWorkMetadata>
</xml>
<-cx:metadata>
<-cx:locator>
<nonSecureIndirect URI=
"http://www.contentguard.com/bookOfJames.spd"/>
</cx:locator>
<-cx:parts>
<-cx:digitalWork licensePartId="chapter1">
<-cx:metadata>
<-xml>
<-cx:simpleDigitalWorkMetadata>
<cx:title>Chapter 1:
In the Beginning</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
</cx:digitalWork>
</cx:parts>
</cx:digitalWork>
</inventory>
<-grant>
<-forall varName="anyone">
</forall>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="chapter1"/>
</grant>
<-forall varName="anyone">
</forall>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="eBook"/>
<-sx:fee>
<-sx:paymentPerUse>
<sx:rate currency="USD">10.00</sx:rate>
</sx:paymentPerUse>
</sx:fee>
</grant>
<-issuer>
<-dsig:Signature>
<-dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=

```

```

"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<-dsig:Reference>
<-dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"
/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbK0QCo941tTeXbjl/Q==</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
allDoedpL...</dsig:SignatureValue>
<-dsig:KeyInfo>
<-dsig:KeyValue>
<-dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
<-details>
<timeOfIssue>2000-01-27T15:30:00</timeOfIssue>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license(uri:"http://www.contentguard.com/bookOfJames.spd")
[
  (canplay(B1,B2,Ch) :-
    Ch=1,
    (canplay(B1,B2,Ch) :-
      Ch>1, get_value(B1,paid,Paid),
      Paid=true, set_value(B1,paid,false,B2)),
    (canpay(B1,B2,I,R) :-
      get_value(B1,issuer_key,I),
      get_value(B1,rate,R),
      set_value(B1,paid,true,B2))
  ],
  (paid=false), (rate=10.00), (currency=usd),
  (issuer_sig="..."), (issuer_key="...") ])

wallet (
[
  (canload(Bw1,Bw2,A) :-
    get_value(Bw1,money,Money),
    N is A+Money,
    set_value(Bw1,money,N,Bw2)),
  (cantansfer(Bw1,Bw2,I,R) :-
    get_value(Bw1,money,Money),
    Money >= R, transfer(I,R),
    N is Money-R,
    set_value(Bw1,money,N,Bw2)) ],
  (money=x) ])

```

### LicenseScript rules:

```

play(Ebook,Chapter) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
<= C |- canplay(B1,B2,Chapter)
pay(Ebook) :
  license(Ebook,C,B1), wallet(Cw,Bw1) ->
  license(Ebook,C,B2), wallet(Cw,Bw2)
<= C |- canpay(B1,B2,I,R),
  Cw |- cantansfer(Bw1,Bw2,I,R)

```

## B.4 Tiered Pricing Model

### XrML code:

```

<-license>
<-inventory>
<-keyHolder licensePartId="issuedToParty">
<-info>
<-dsig:KeyValue>

```

```

<-dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<-cx:digitalWork licensePartId="eBook">
<-cx:locator>
<nonSecureIndirect
URI="http://www.contentguard.com/sampleBook.spd"/>
</cx:locator>
</cx:digitalWork>
<-sx:trackReport licensePartId="trackPrint">
<-sx:stateReference>
<-uddi>
<-serviceKey>
<uuid>1F8903B0-FC03-4c5b-A445-AAFCCEC01111</uuid>
</serviceKey>
</uddi>
<-serviceParameters>
<-datum>
<cx:print/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:trackReport>
</inventory>
<-grant>
<-keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
<cx:print/>
<-cx:digitalWork licensePartIdRef="eBook"/>
<-allConditions>
<sx:trackReport licensePartIdRef="trackPrint"/>
<-sx:trackQuery>
<-sx:stateReference>
<-uddi>
<-serviceKey>
<uuid>1F8903B0-FC03-4c5b-A445-AAFCCEC01111</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
<sx:notMoreThan>9</sx:notMoreThan>
</sx:trackQuery>
<-sx:fee>
<-sx:paymentPerUse>
<sx:rate currency="USD">10.00</sx:rate>
</sx:paymentPerUse>
<-sx:to>
<-sx:aba>
<sx:institution>139371581</sx:institution>
<sx:account>111111</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</allConditions>
</grant>
<-grant>
<-keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
<cx:print/>
<cx:digitalWork licensePartIdRef="eBook"/>
<-allConditions>
<sx:trackReport licensePartIdRef="trackPrint"/>
<-sx:trackQuery>
<-sx:stateReference>
<-uddi>
<-serviceKey>
<uuid>1F8903B0-FC03-4c5b-A445-AAFCCEC01111</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
<sx:notLessThan>10</sx:notLessThan>
</sx:trackQuery>
<-sx:fee>
<-sx:paymentPerUse>
<sx:rate currency="USD">8.00</sx:rate>
</sx:paymentPerUse>
<-sx:to>
<-sx:aba>

```

```

<sx:institution>139371581</sx:institution>
<sx:account>111111</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</allConditions>
</grant>
</license>

```

### LicenseScript code:

```

license(url="http://www.contentguard.com/sampleBook.spd"
[ (canprint(B1,B2) :-
  get_value(B1,paid,Paid),
  Paid=true,
  get_value(B1,prints,Prints),
  set_value(B1,prints,Prints+1,B2),
  set_value(B1,paid,false,B2),
  get_value(B1,skey,K),
  uddi_trackprint(K,print)),
  (canpay(B1,B2,H,K,I,A,R) :-
  get_value(B1,prints,Prints),
  Prints<=10,
  get_value(B1,rate1,R),
  get_value(B1,institution,I),
  get_value(B1,account,A),
  get_value(B1,keyholder,H),
  get_value(B1,skey,K),
  set_value(B1,paid,true,B2)),
  (canpay(B1,B2,H,K,I,A,R) :-
  get_value(B1,prints,Prints),
  Prints>10,
  get_value(B1,rate2,R),
  get_value(B1,institution,I),
  get_value(B1,account,A),
  get_value(B1,keyholder,H),
  get_value(B1,skey,K),
  set_value(B1,paid,true,B2))
],
[ (paid=false), (rate1=10),
  (rate2=8), (prints=0),
  (keyholder="<mod>Efgao6NYf...<exp>AQAQAA=="
  (skey="1F8903B0-FC03-4c5b-A445-AAFCCEC01111"),
  (institution="139371581"),
  (account="1111111" )]
)
wallet (
[ (canload(Bw1,Bw2,A) :-
  get_value(Bw1,money,Money),
  N is A+Money,
  set_value(Bw1,money,N,Bw2)),
  (cantransfer(Bw1,Bw2,H,K,I,A,R) :-
  get_value(Bw1,money,Money),
  Money >= R,
  uddi_transfer(H,I,A,R),
  N is Money-R,
  set_value(Bw1,money,N,Bw2)) ],
[ (money=x) ])

```

### LicenseScript rules:

```

print(Ebook,Chapter) :
  license(Ebook,C,B1) -> license(Ebook,C,B2)
<= C |- canplay(B1,B2,Chapter)
pay(Ebook) :
  license(Ebook,C,B1), wallet(Cw,Bw1) ->
  license(Ebook,C,B2), wallet(Cw,Bw2)
<= C |- canpay(B1,B2,H,K,I,A,R),
  Cw |- cantransfer(Bw1,Bw2,H,K,I,A,R)

```

## B.5 Pay per View/User

### XrML code:

```

-<license>
-<grant>
-<forAll varName="anyone">

```

```

</forAll>
<principal varRef="anyone"/>
<cx:play/>
-<cx:digitalWork>
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Air Force One</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
<secureIndirect
URI="http://sonyPictures.com/AirForceOne">
<dsig:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
0kcc24a3zFW9OPT1qEIqSg==</dsig:DigestValue>
</secureIndirect>
</cx:locator>
</cx:digitalWork>
-<sx:fee>
-<sx:paymentPerUse>
<sx:rate currency="USD">2.00</sx:rate>
</sx:paymentPerUse>
-<sx:to>
-<sx:aba>
<sx:institution>139371581</sx:institution>
<sx:account>111111</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</grant>
</license>

```

### LicenseScript code:

```

license(url:"http://sonyPictures.com/AirForceOne",
[ (canplay(B1,B2) :-
  get_value(B1,paid,Paid),
  Paid=true,
  set_value(B1,paid,false,B2)),
  (canpay(B1,B2,I,A,R) :-
  get_value(B1,prints,Prints),
  Prints>10,
  get_value(B1,rate,R),
  get_value(B1,institution,I),
  get_value(B1,account,A),
  set_value(B1,paid,true,B2))
],
[ (digest="0kcc24a3zFW9OPT1qEIqSg=="
  (rate=2.00), (currency=usd), (paid=false),
  (institution=139371581), (account=111111)
)
)
wallet (
[ (canload(Bw1,Bw2,A) :-
  get_value(Bw1,money,Money),
  N is A+Money,
  set_value(Bw1,money,N,Bw2)),
  (cantransfer(Bw1,Bw2,I,A,R) :-
  get_value(Bw1,money,Money),
  Money >= R,
  transfer(I,A,R),
  N is Money-R,
  set_value(Bw1,money,N,Bw2)) ],
[ (money=x) ])

```

### LicenseScript rules:

```

play(Movie) :
  license(Movie,C,B1) -> license(Movie,C,B2)
<= C |- canplay(B1,B2)
pay(Movie) :
  license(Movie,C,B1), wallet(Cw,Bw1) ->
  license(Movie,C,B2), wallet(Cw,Bw2)
<= C |- canpay(B1,B2,I,A,R),
  Cw |- cantransfer(Bw1,Bw2,I,A,R)

```

## B.6 Subscription

XrML code:

```
<license>
<title>A Subscription Offer</title>
<grant>
<forall varName="anyone">
</forall>
<forall varName="oneYear">
<library:validityIntervalDurationPattern>
<library:duration>P1Y</library:duration>
</library:validityIntervalDurationPattern>
</forall>
<keyHolder varRef="anyone">
</keyHolder>
<obtain/>
<grant>
<keyHolder varRef="anyone">
</keyHolder>
<possessProperty/>
<ebook:subscription/>
<validityInterval varRef="oneYear"/>
</grant>
<sx:fee>
<sx:paymentPerUse>
<sx:rate>100.00</sx:rate>
</sx:paymentPerUse>
</sx:fee>
</grant>
<forall varName="anyone">
</forall>
<forall varName="oneYear">
<library:validityIntervalDurationPattern>
<library:duration>P1M</library:duration>
</library:validityIntervalDurationPattern>
</forall>
<keyHolder varRef="anyone">
</keyHolder>
<obtain/>
<grant>
<keyHolder varRef="anyone">
</keyHolder>
<possessProperty/>
<ebook:subscription/>
<validityInterval varRef="oneYear"/>
</grant>
<sx:fee>
<sx:paymentPerUse>
<sx:rate>10.00</sx:rate>
</sx:paymentPerUse>
</sx:fee>
</grant>
<issuer>
<dsig:Signature>
<dsig:SignedInfo>
<dsig:CanonicalizationMethod
Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod
Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<dsig:Reference>
<dsig:Transforms>
<dsig:Transform
Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCo941tTEbj1/Q==</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
zIRYaxl5E...
</dsig:SignatureValue>
<dsig:KeyInfo>
<dsig:KeyValue>
<dsig:RSAKeyValue>
```

```
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
<details>
<timeOfIssue>2001-01-27T15:30:00</timeOfIssue>
<validityInterval>
<notBefore>2000-02-03T17:26:00</notBefore>
<notAfter>3000-02-03T17:26:00</notAfter>
</validityInterval>
</details>
</issuer>
</license>
```

LicenseScript code:

```
offer (ebook,
[ (cansubscribe (B1, B2, U) :-
  get_value (B1, notbefore, NB),
  get_value (B1, notafter, NA),
  today (D), D > NA, D < NB,
  set_value (B1, expires, D, B2),
  set_value (B1, userid, U, B2)),
  (canactivate (B1, B2, U) :-
  get_value (B1, paid, Paid), Paid = true,
  get_value (B1, userid, User), U = User,
  set_value (B1, activated, true, B2)),
  (canpay (B1, B2, U, I, R, T) :-
  T = subscribe,
  get_value (B1, subrate, R),
  get_value (B1, userid, User), U = User,
  get_value (B1, issuer_key, I),
  set_value (B1, paid, true, B2)),
  (canpay (B1, B2, U, I, R, T) :-
  T = use,
  get_value (B1, userid, User), U = User,
  get_value (B1, ratepuse, R),
  get_value (B1, issuer_key, I),
  set_value (B1, paid, true, B2))
]),
[ (notbefore="2000-02-03T17:26:00"),
  (notafter="3000-02-03T17:26:00"),
  (issuedtime="2001-01-27T15:30:00"),
  (issuer_key="<mod>g8NRYMG30...<exp>AQABAA=="),
  (digest="PB4QbKQCo941tTEbj1/Q="),
  (issuer_signature="zIRYaxl5E..."),
  (expires=nil), (userid=nil),
  (subrate=100.00), (ratepuse=10.00),
  (paid=false), (activated=false)
])
wallet (
[ (canload (Bw1, Bw2, A) :-
  get_value (Bw1, money, Money),
  N is A+Money,
  set_value (Bw1, money, N, Bw2)),
  (cantransfer (Bw1, Bw2, I, R) :-
  get_value (Bw1, money, Money),
  Money >= R, transfer (I, R),
  N is Money-R,
  set_value (Bw1, money, N, Bw2)) ],
[ (money=x) ])
```

LicenseScript rules:

```
subscribe (Service, User) :
  offer (Service, C, B1) ->
  offer (Service, C, B1), license (Service, C, B2)
<= C |- cansubscribe (B1, B2, B3, User)
pay (Service, User, Type) :
  license (Service, C, B1), wallet (Cw, Bw1) ->
  license (Service, C, B2), wallet (Cw, Bw2)
<= C |- canpay (B1, B2, User, I, R, Type),
  Cw |- cantransfer (Bw1, Bw2, I, R)
activate (Service, User) :
  license (Service, C, B1) -> license (Service, C, B2)
<= C |- canactivate (B1, B2)
```



## B.7 Territory Restriction

XrML code:

```
<license>
<inventory>
<cx:digitalWork licensePartId="eBook">
<cx:metadata>
<xml>
<cx:simpleDigitalWorkMetadata>
<cx:title>XrML Unleashed</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
<cx:locator>
<cx:nonSecureIndirect
URI="http://www.contentguard.com/xrmlUnleashed.spd"/>
</cx:locator>
</cx:digitalWork>
</inventory>
<grant>
<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="eBook"/>
<allConditions>
<sx:territory>
<sx:location>
<sx:country>US</sx:country>
</sx:location>
</sx:territory>
<sx:fee>
<sx:paymentFlat>
<sx:rate currency="USD">5.00</sx:rate>
<sx:paymentRecord>
<sx:stateReference>
<uddi>
<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
<serviceParameters>
<datum>
<principal varRef="anyone"/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
</allConditions>
</grant>
<grant>
<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="eBook"/>
<sx:fee>
<sx:paymentFlat>
<sx:rate currency="USD">7.00</sx:rate>
<sx:paymentRecord>
<sx:stateReference>
<uddi>
<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C21111</uuid>
</serviceKey>
</uddi>
<serviceParameters>
<datum>
<principal varRef="anyone"/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
</grant>
</license>
```

LicenseScript code:

```
license(uri="http://www.contentguard.com/xrmlUnleashed.spd",
[ (canplay(B1,B2) :-
  get_value(B1,paid,Paid), Paid=true),
  (canpay(B1,B2,R,K) :-
    identify(Country), get_value(B1,country,Cnt),
    Country=Cnt, get_value(B1,rate1,R),
    get_value(B1,skey,K),
    set_value(B1,paid,true,B2)),
  (canpay(B1,B2,R,K) :-
    identify(Country), get_value(B1,country,Cnt),
    Country=\Cnt, get_value(B1,rate2,R),
    get_value(B1,skey,K),
    set_value(B1,paid,true,B2))
]),
[ (country=us), (rate1=5.00), (rate2=7.00), (paid=false),
  (skey="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
])
wallet(
[ (canload(Bw1,Bw2,A) :-
  get_value(Bw1,money,Money),
  N is A+Money,
  set_value(Bw1,money,N,Bw2)),
  (cantransfer(Bw1,Bw2,K,R) :-
    get_value(Bw1,money,Money),
    Money >= R, transfer(K,R),
    track(K,R), N is Money-R,
    set_value(Bw1,money,N,Bw2)) ],
[ (money=x) ])
```

LicenseScript rules:

```
play(Meta) :
  license(Meta,C,B1) -> license(Movie,C,B2)
  <= C |- canplay(B1,B2)
pay(Meta) :
  license(Meta,C,B1), wallet(Cw,Bw1) ->
  license(Meta,C,B2), wallet(Cw,Bw2)
  <= C |- canpay(B1,B2,R,K),
  Cw |- cantransfer(Bw1,Bw2,R,K)
```

## B.8 Temporal Ordering of Exercising Rights

XrML code:

```
<license>
<inventory>
+<keyHolder licensePartId="Alice">
<digitalResource licensePartId="Commercial">
<cx:metadata>
<xml>
<cx:simpleDigitalWorkMetadata>
<cx:title>Toyota Ad</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
</digitalResource>
<digitalResource licensePartId="music">
<dsig:Reference URI=
"http://www.server.com/downloads/anInterestingSong.mp3">
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
qZk+NkcGgWq6PiVxeFDCbJzQ2J0=
</dsig:DigestValue>
</dsig:Reference>
</digitalResource>
</inventory>
<grant>
<keyHolder licensePartIdRef="Alice">
</keyHolder>
<cx:play/>
<cx:digitalWork licensePartIdRef="Commercial"/>
<sx:trackReport>
<sx:stateReference licensePartId="AdState">
```

```

-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
<sx:communicationFailurePolicy>
lax
</sx:communicationFailurePolicy>
</sx:trackReport>
</grant>
-<grant>
-<keyHolder licensePartIdRef="Alice">
</keyHolder>
<cx:play/>
<cx:digitalWork licensePartIdRef="Music"/>
<!-- trackQuery -->
<!-- stateReference licensePartId="AdState" -->
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
<!-- notLessThan>1</notLessThan -->
<!-- trackQuery -->
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCo941tTExbj1/Q==</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
zIRYaxl5E...</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
</details>
<timeOfIssue>2001-01-27T15:30:00</timeOfIssue>
<!-- validityInterval -->
<notBefore>2000-02-03T17:26:00</notBefore>
<notAfter>3000-02-03T17:26:00</notAfter>
</validityInterval>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (
http://www.server.com/downloads/anInterestingSong.mp3,
[ (canlisten (B1,B2,H) :-
H>1, get_value (B1,skey,Skey),
track_query (Skey)),
],
[ (digest="qZk+NkcGgWq6PiVxeFDcbJzQ2J0="),
(user=alice),
(skey="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
])
license (some_advertisements,

```

```

[ (canlisten (B1,B2,NH) :-
get_value (B1,heard,H), NH is H+1,
set_value (B1,heard,NH,B2))
],
[ (heard=0) ]])

```

### LicenseScript rules:

```

listen (Ads,Music) :
license (Ads,Ca,Ba1), license (Music,Cm,Bm1) ->
license (Ads,Ca,Ba2), license (Music,Cm,Bm2)
<= Ca |- canlisten (Ba1,Ba2,NH),
Cm |- canlisten (Bm1,Bm2,H)

```

## B.9 Component Based Model

### XrML code:

```

-<license>
-<inventory>
-<cx:digitalWork licensePartId="WebSite">
<cx:description xml:lang="en">
A very good Web site</cx:description>
<cx:description xml:lang="fr">
Un tres bon Web site</cx:description>
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Web site for the Book of James</cx:title>
<cx:creator>James the first</cx:creator>
<cx:owner>Mike the man</cx:owner>
<cx:copyright>
Copyright 1999 Harper Collins Publishers. All Rights Reserved.
</cx:copyright>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
<nonSecureIndirect URI="http://www.somedomain.com/bookOfJames"/>
</cx:locator>
-<cx:parts>
-<cx:digitalWork licensePartId="webPageOne">
<cx:description>
Web page containing text and images that make up
Chapter 1 for the Book of James</cx:description>
-<cx:locator>
<nonSecureIndirect
URI="http://www.somedomain.com/bookOfJames/chap1.htm"/>
</cx:locator>
</cx:digitalWork>
-<cx:digitalWork licensePartId="webPageOneImage1">
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Image 1: Photon Celebrity Dogs</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
<nonSecureIndirect URI=
"http://www.somedomain.com/bookOfJames/images/chap1Image1.jpg"/>
</cx:locator>
</cx:digitalWork>
</cx:parts>
</cx:digitalWork>
</inventory>
-<grant>
-<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="webPageOne"/>
-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">1.00</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>

```

```

-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
-<serviceParameters>
-<datum>
<principal varRef="anyone"/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
</grant>
-<grant>
-<forall varName="anyone">
</forall>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="webPageOneImage1"/>
-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">0.20</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-CRT882MAOFH2</uuid>
</serviceKey>
</uddi>
-<serviceParameters>
-<datum>
<principal varRef="anyone"/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbK0QCo94ltTExbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>alIDoedL...</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
-<details>
<timeOfIssue>2000-01-27T15:30:00</timeOfIssue>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (http://www.somedomain.com/bookOfJames/,
[ (canreuse (B1, B2, Obj) :-
    type_of (Obj, Typ), Typ=htm,

```

```

    get_value (B1, paid1, Paid), Paid=true,
    get_value (B1, objs, Objs),
    member (Obj, Objs),
    remove (B1, objs, B2),
    remove (B1, paid2, B2),
    remove (B1, rate2, B2)),
    (canreuse (B1, B2, Obj) :-
        type_of (Obj, Typ), Typ=jpg,
        get_value (B1, paid2, Paid), Paid=true,
        get_value (B1, objs, Objs),
        member (Obj, Objs),
        remove (B1, objs, B2),
        remove (B1, paid1, B2),
        remove (B1, rate1, B2)),
    (canpay (B1, B2, Obj, R, K) :-
        type_of (Obj, Typ), Typ=htm,
        get_value (B1, rate1, R),
        set_value (B1, paid1, true, B2),
        get_value (B1, skey1, K)),
    (canpay (B1, B2, Obj, R, K) :-
        type_of (Obj, Typ), Typ=jpg,
        get_value (B1, rate2, R),
        set_value (B1, paid2, true, B2),
        get_value (B1, skey2, K))
    ],
[ (creator="james-the-first"),
  (type1=htm), (type2=jpg),
  (objs=["../chap1.htm", "../chap1Image1.jpg"]),
  (skey1="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
  (skey2="D04951E4-332C-4693-B7DB-CRT882MAOFH2"),
  (rate1=1.00), (rate2=0.20),
  (paid1=false), (paid2=false)
])
wallet (
[ (canload (Bw1, Bw2, A) :-
    get_value (Bw1, money, Money),
    N is A+Money,
    set_value (Bw1, money, N, Bw2)),
  (cantransfer (Bw1, Bw2, K, R) :-
    get_value (Bw1, money, Money),
    Money >= R, transfer (K, R),
    uddi_trackpay (K, R),
    track (K, R), N is Money-R,
    set_value (Bw1, money, N, Bw2)) ],
[ (money=x) ])

```

### LicenseScript rules:

```

license (Web, C, B1) ->
license (Web, C, B1), license (Obj, C, B2)
<= C |- canreuse (B1, B2, Obj)
pay (Obj) :
license (Obj, C, B1), wallet (Cw, Bw1) ->
license (Obj, C, B2), wallet (Cw, Bw2)
<= C |- canpay (B1, B2, Obj, R, K),
Cw |- cantransfer (Bw1, Bw2, K, R)

```

## B.10 User Type Based Model

### XrML code:

```

-<licenseGroup>
-<license>
-<grant>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6Nyf...</dsig:Modulus>
<dsig:Exponent>AQAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<possessProperty/>
<sony:sonyClubMember/>

```

```

</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"*/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCCo941tTExbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
AYmqOhSHb...
</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>X0j9q90Ax...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
</issuer>
</license>
-<license>
-<inventory>
-<cx:digitalWork licensePartId="movie">
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Air Force One</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
-<secureIndirect URI=
"http://sonyPictures.com/AirForceOne">
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
0kccZ4a3zFW9OPT1qEIqSg==
</dsig:DigestValue>
</secureIndirect>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grant>
-<forall varName="anyone">
</forall>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartId="movie"/>
-<allConditions>
-<prerequisiteRight>
<principal varRef="anyone"/>
<possessProperty/>
<sony:sonyClubMember/>
-<trustedIssuer>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>X0j9q90Ax...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
</trustedIssuer>
</prerequisiteRight>
-<sx:fee>
-<sx:paymentPerUse>
<sx:rate currency="USD">5.00</sx:rate>
</sx:paymentPerUse>
-<sx:to>
-<sx:aba>
<sx:institution>139371581</sx:institution>
<sx:account>111111</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</allConditions>
</grant>
-<forall varName="anyone">
</forall>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartId="movie"/>
-<sx:fee>
-<sx:paymentPerUse>
<sx:rate currency="USD">7.00</sx:rate>
</sx:paymentPerUse>
-<sx:to>
-<sx:aba>
<sx:institution>139371581</sx:institution>
<sx:account>111111</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</grant>
</license>
</licenseGroup>

```

### LicenseScript code:

```

license (http://sonyPictures.com/AirForceOne,
[
  (canplay (B1, B2, U) :-
    get_value (B1, paid, Paid), Paid=true,
    set_value (B1, paid, false, B2)),
  (canpay (B1, B2, U, R, TI) :-
    get_value (B1, trustedissuer, TI),
    get_value (B1, keyholder, KH),
    is_member (U, TI, KH),
    get_value (B1, peruse1, R),
    set_value (B1, paid, true, B2)),
  (canpay (B1, B2, U, R, TI) :-
    get_value (B1, trustedissuer, TI),
    get_value (B1, keyholder, KH),
    not (is_member (U, TI, KH)),
    get_value (B1, peruse2, R),
    set_value (B1, paid, true, B2))
]),
[
  (issuer_sig="AYmqOhSHb...")
  (keyholder="<mod>Efgao6NYf...<exp>AQQA=="),
  (digest="0kccZ4a3zFW9OPT1qEIqSg=="),
  (peruse1=5.00), (peruse2=7.00), (paid=false),
  (institution="139371581"),
  (account="111111"),
  (trustedissuer="<mod>X0j9q90Ax...<exp>AQABAA==")
])
]

wallet (
[
  (canload (Bw1, Bw2, A) :-
    get_value (Bw1, money, Money),
    N is A+Money,
    set_value (Bw1, money, N, Bw2)),
  (canttransfer (Bw1, Bw2, I, R) :-
    get_value (Bw1, money, Money),
    Money >= R, transfer (I, R),
    N is Money-R,
    set_value (Bw1, money, N, Bw2)) ],
[
  (money=x) ])

LicenseScript rules:

play (Movie) :
  license (Movie, C, B1) -> license (Movie, C, B2)
  <= C | - canplay (B1, B2)
pay (Obj) :
```

```

license (Obj,C,B1), wallet (Cw,Bw1) ->
license (Obj,C,B2), wallet (Cw,Bw2)
<= C |- canpay (B1,B2,U,R,TI),
    Cw |- cantransfer (Bw1,Bw2,TI,R)

```

```

get_value (B1,skey,K),
set_value (B1,paid,true,B2)
],
[ (paid=false), (rate=2.00),
  (skey="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
  (trustedissuer="..."),
]
)

```

## B.11 Site Licenses

XrML code:

```

-<license>
-<grant>
-<forAll varName="anyone">
</forAll>
-<forAll varName="anySiteContent">
<xmlExpression>
--Expression that matches all site content--
</xmlExpression>
</forAll>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork varRef="anySiteContent"/>
-<prerequisiteRight>
<principal varRef="anyone"/>
<possessProperty/>
-<acn:siteLicense>
<acn:typeId>promotion710</acn:typeId>
<acn:url>www.somedomain.com/somesite</acn:url>
</acn:siteLicense>
-<trustedIssuer>
+<keyHolder>
</trustedIssuer>
</prerequisiteRight>
</grant>
-<grant>
-<forAll varName="anySiteContent">
<xmlExpression>
--XPath expression matching any
piece of content on www.somedomain.com/somesite--
</xmlExpressi
</forAll>
-<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork varRef="anySiteContent"/>
-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">2.00</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>
D04951E4-332C-4693-B7DB-D3D1D1C20844
</uuid>
</serviceKey>
</uddi>
-<serviceParameters>
-<datum>
<cx:digitalWork varRef="anySiteContent"/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
</grant>
</license>

```

LicenseScript code:

```

license (www.somedomain.com/somesite,
[ (canplay (B1,B2) :-
  get_value (B1,paid,Paid),
  Paid=true),
  (canpay (B1,B2,TI,R,K) :-
  get_value (B1,trustedissuer,TI),
  get_value (B1,rate,R),

```

```

wallet (
[ (canload (Bw1,Bw2,A) :-
  get_value (Bw1,money,Money),
  N is A+Money,
  set_value (Bw1,money,N,Bw2)),
  (cantransfer (Bw1,Bw2,TI,R,K) :-
  get_value (Bw1,money,Money),
  Money >= R, transfer (TI,R,K),
  N is Money-R,
  set_value (Bw1,money,N,Bw2)) ],
[ (money=x) ])

```

LicenseScript rules:

```

play (Content) :
  license (Content,C,B1) -> license (Content,C,B2)
  <= C |- canplay (B1,B2)
pay (Obj) :
  license (Obj,C,B1), wallet (Cw,Bw1) ->
  license (Obj,C,B2), wallet (Cw,Bw2)
  <= C |- canpay (B1,B2,TI,R,K),
  Cw |- cantransfer (Bw1,Bw2,TI,R,K)

```

## B.12 Payment to Multiple Rights Holders

XrML code:

```

-<license>
-<grant>
-<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
<cx:print/>
-<cx:digitalWork>
-<cx:locator>
<nonSecureIndirect
URI="http://www.contentguard.com/sampleBook.spd"/>
</cx:locator>
</cx:digitalWork>
-<allConditions>
-<sx:fee>
-<sx:paymentPerUse>
<sx:rate currency="USD">0.10</sx:rate>
</sx:paymentPerUse>
-<sx:to>
-<sx:aba>
<sx:institution>222371863</sx:institution>
<sx:account>123456</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
-<sx:fee>
-<sx:paymentPerUse>
<sx:rate currency="USD">0.05</sx:rate>
</sx:paymentPerUse>
-<sx:to>
-<sx:aba>
<sx:institution>139371581</sx:institution>
<sx:account>111111</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</allConditions>
</grant>
</license>

```

LicenseScript code:

```

license (http://www.contentguard.com/sampleBook.spd,
[ (canprint (B1, B2) :-
  get_value (B1, paid, Paid), Paid=true,
  set_value (B1, paid, false, B2)),
  (canpay (B1, B2, R1, I1, A1, R2, I2, A2) :-
  get_value (B1, ratel, R1),
  get_value (B1, institutel, I1),
  get_value (B1, account1, A1),
  get_value (B1, rate2, R2),
  get_value (B1, institutel, I2),
  get_value (B1, account2, A2),
  set_value (B1, paid, true, B2))
],
[ (paid=false), (ratel=0.10),
  (institutel="222371863"),
  (account1="123456"),
  (rate2=0.50), (institute2="139371581"),
  (account2="111111")
])
wallet (
[ (canload (Bw1, Bw2, A) :-
  get_value (Bw1, money, Money),
  N is A+Money,
  set_value (Bw1, money, N, Bw2)),
  (cantransfer (Bw1, Bw2, R1, I1, A1, R2, I2, A2) :-
  get_value (Bw1, money, Money),
  Money >= R1+R2,
  transfer (R1, I1, A1),
  transfer (R2, I2, A2),
  N is Money-(R1+R2),
  set_value (Bw1, money, N, Bw2)) ],
[ (money=x) ])

```

### LicenseScript rules:

```

print (Ebook) :
  license (Ebook, C, B1) -> license (Ebook, C, B2)
<= C |- canprint (B1, B2)
pay (Ebook) :
  license (Ebook, C, B1), wallet (Cw, Bw1) ->
  license (Ebook, C, B2), wallet (Cw, Bw2)
<= C |- canpay (B1, B2, R1, I1, A1, R2, I2, A2),
  Cw |- cantransfer (Bw1, Bw2, R1, I1, A1, R2, I2, A2)

```

## B.13 Personal Lending

### XrML code:

```

-<license>
-<inventory>
-<cx:digitalWork licensePartId="dvdMovie">
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Air Force One</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
-<secureIndirect URI="http://sonyPictures.com/AirForceOne">
<dsig:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
0kccZ4a3zFW9OPTIqEIqSg==</dsig:DigestValue>
</secureIndirect>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grantGroup>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQAQA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>

```

```

</info>
</keyHolder>
-<validityInterval licensePartId="rentalPeriod">
<notBefore>2001-05-20T19:28:00</notBefore>
<notAfter>2001-05-29T19:28:00</notAfter>
</validityInterval>
-<grant>
<cx:play/>
<cx:digitalWork licensePartIdRef="dvdMovie"/>
-<sx:seekApproval>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:seekApproval>
</grant>
-<grantGroup>
-<grant>
<cx:loan/>
<cx:digitalWork licensePartIdRef="dvdMovie"/>
</grant>
-<grant>
<forAll varName="anyone">
</forAll>
-<forAll varName="loanPeriod">
</forAll>
<issue/>
-<grant>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="dvdMovie"/>
-<allConditions>
<validityInterval licensePartIdRef="rentalPeriod"/>
<validityInterval varRef="loanPeriod"/>
</allConditions>
</grant>
-<sx:trackReport>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
-<serviceParameters>
-<datum>
<validityInterval varRef="loanPeriod"/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:trackReport>
</grant>
</grantGroup>
</license>

```

LicenseScript code and rules are provided in section 4.2.1.

## B.14 Giving

### XrML code:

```

-<license>
-<inventory>
-<cx:digitalWork licensePartId="dvdMovie">
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Air Force One</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
-<secureIndirect URI=
"http://sonyPictures.com/AirForceOne">

```

```

<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
0kccZ4a3zFW9OPT1qEIqSg==
</dsig:DigestValue>
</secureIndirect>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grantGroup>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQQA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">10.00</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>
D04951E4-332C-4693-B7DB-D3D1D1C20844
</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
-<grant>
<cx:play/>
<cx:digitalWork licensePartIdRef="dvdMovie"/>
-<sx:seekApproval>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>
D04951E4-332C-4693-B7DB-D3D1D1C20844
</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:seekApproval>
</grant>
-<grantGroup>
-<grant>
<cx:transfer/>
<cx:digitalWork licensePartIdRef="dvdMovie"/>
</grant>
-<grant>
-<forAll varName="anyone">
</forAll>
<issue/>
-<grant>
<principal varRef="anyone"/>
<cx:play/>
<cx:digitalWork licensePartIdRef="dvdMovie"/>
</grant>
-<sx:trackReport>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>
D04951E4-332C-4693-B7DB-D3D1D1C20844
</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:trackReport>
</grant>
</grantGroup>
</grantGroup>
</license>

```

## LicenseScript code:

```

license (dvdmovie:"http://sonyPictures.com/AirForceOne",
[ (canplay (B1,B2) :-
  get_value (B1,skey,K),
  uddi_seekApproval (K)),
  (cantransfer (B1,B2,U) :-
  get_value (B1,paid,Paid),
  Paid=true,
  set_value (B1,keyholder,U,B2),
  get_value (B1,skey,K),
  track_report (K)
  (canpay (B1,B2,R,K) :-
  get_value (B1,rate,R),
  get_value (B1,skey,K),
  set_value (B1,paid,true,B2)
  )
  ]),
[ (digest="0kccZ4a3zFW9OPT1qEIqSg=="),
  (keyholder="<mod>Efgao6NYf...<exp>AQQA=="),
  (rate=10.00), (paid=false),
  (skey="D04951E4-332C-4693-B7DB-D3D1D1C20844")
  ]
]
wallet (
[ (canload (Bw1,Bw2,A) :-
  get_value (Bw1,money,Money),
  N is A+Money,
  set_value (Bw1,money,N,Bw2)),
  (cantransfer (Bw1,Bw2,R,K) :-
  get_value (Bw1,money,Money),
  Money >= R, uddi_transfer (R,K),
  N is Money-R,
  set_value (Bw1,money,N,Bw2)) ],
[ (money=x) ])

```

## LicenseScript rules:

```

play (Movie) :
  license (Movie,C,B1) -> license (Movie,C,B2)
  <= C |- canplay (B1,B2)
transfer (Movie,User) :
  license (Movie,C,B1) -> license (Movie,C,B2)
  <= C |- cantransfer (B1,B2,User)
pay (Ebook) :
  license (Ebook,C,B1), wallet (Cw,Bw1) ->
  license (Ebook,C,B2), wallet (Cw,Bw2)
  <= C |- canpay (B1,B2,R,K),
  Cw |- cantransfer (Bw1,Bw2,R,K)

```

## B.15 Superdistribution

### XrML code:

```

-<license>
-<inventory>
-<keyHolder licensePartId="Alice">
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>p8sn4KeeR...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
-<cx:digitalWork licensePartId="eBook">
-<cx:locator>
<nonSecureIndirect
URI="http://www.contentguard.com/xrmlUnleashed.spd"/>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grantGroup>
-<keyHolder licensePartIdRef="Alice">
</keyHolder>
-<grant>
<cx:play/>
<cx:digitalWork licensePartIdRef="eBook"/>

```

```

-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">2.00</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
-<serviceParameters>
-<datum>
-<keyHolder licensePartIdRef="Alice">
</keyHolder>
</datum>
-<datum>
<cx:digitalWork licensePartIdRef="eBook"/>
</datum>
-<datum>
<cx:play/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
-<sx:to>
-<sx:aba>
<sx:institution>222371864</sx:institution>
<sx:account>123457</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</grant>
-<grant>
<cx:print/>
<cx:digitalWork licensePartIdRef="eBook"/>
-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">15.00</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
-<serviceParameters>
-<datum>
-<keyHolder licensePartIdRef="Alice">
</keyHolder>
</datum>
-<datum>
<cx:digitalWork licensePartIdRef="eBook"/>
</datum>
-<datum>
<cx:print/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
-<sx:to>
-<sx:aba>
<sx:institution>222371864</sx:institution>
<sx:account>123457</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</grant>
</grantGroup>
-<grant>
-<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
<obtain/>
-<grantGroup licensePartId="superdistributedGrants">
-<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
</grant>
<cx:play/>
<cx:digitalWork licensePartIdRef="eBook"/>
-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">2.00</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
-<serviceParameters>
-<datum>
<cx:play/>
</datum>
-<datum>
<cx:digitalWork licensePartIdRef="eBook"/>
</datum>
-<datum>
<principal varRef="anyone"/>
</datum>
</serviceParameters>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
-<sx:to>
-<sx:aba>
<sx:institution>222371864</sx:institution>
<sx:account>123457</sx:account>
</sx:aba>
</sx:to>
</grant>
</grantGroup>
-<grant>
-<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
<obtain/>
-<grantGroup licensePartId="superdistributedGrants">
-<forAll varName="anyone">
</forAll>
<principal varRef="anyone"/>
</grant>
<cx:play/>

```



```

</sx:fee>
</grant>
</grantGroup>
</grant>
</license>

```

LicenseScript code and rules are provided in section 4.2.2.

## B.16 Distributor Copies

XrML code:

```

-<licenseGroup>
-<license>
-<grant>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQAQAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<possessProperty/>
<murphy:distributor/>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCCo941tTEbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
aMCJIHO2g...
</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>p8sN4KeeR...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
</issuer>
</license>
-<license>
-<inventory>
-<keyHolder licensePartId="issuedToParty">
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQAQAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
</inventory>
-<grant>
-<forAll varName="anyone">
</forAll>

```

```

-<keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
<issue/>
-<grant>
<principal varRef="anyone"/>
<cx:play/>
-<digitalResource>
-<dsig:Reference URI=
"http://www.server.com/downloads/anInterestingSong.mp3">
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
qZk+NkcGgWq6PiVxeFDCbJzQ2J0=
</dsig:DigestValue>
</dsig:Reference>
</digitalResource>
</grant>
-<allConditions>
-<sx:exerciseLimit>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuiid>
D04951E4-332C-4693-B7DB-D3D1D1C20844
</uuiid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:exerciseLimit>
-<validityInterval>
<notBefore>2001-05-25T00:00:00</notBefore>
<notAfter>2002-05-25T00:00:00</notAfter>
</validityInterval>
-<prerequisiteRight>
-<keyHolder licensePartIdRef="issuedToParty">
</keyHolder>
<possessProperty/>
<murphy:distributor/>
-<trustedIssuer>
-<keyHolder>
-<info>
-<dsig:KeyValue>
<dsig:RSAKeyValue>
<dsig:Modulus>p8sN4KeeR...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
</trustedIssuer>
</prerequisiteRight>
</allConditions>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCCo941tTEbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
mFdJfMcnl...
</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>

```

```

</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
-<details>
<timeOfIssue>2001-05-24T14:56:00</timeOfIssue>
-<validityInterval>
<notBefore>2001-05-25T00:00:00</notBefore>
<notAfter>2003-05-25T00:00:00</notAfter>
</validityInterval>
</details>
</issuer>
</license>
</licenseGroup>

```

### LicenseScript code:

```

license (
http://www.server.com/downloads/anInterestingSong.mp3,
[ (canplay (B1,B2,U) :-
  get_value (B1,Skey,K) ,
  seek_approval (K,U) ) ,
  (canissue (B1,B2,B3,I) :-
  get_value (B1,trustedissuer,Issuer) ,
  I=Issuer, today (D) ,
  get_value (B1,notbefore,NB) ,
  get_value (B1,notafter,NA) ,
  D>NB, D<NA, get_value (B1,issued,Issued) ,
  get_value (B1,max,Max) , Issued<=Max,
  set_value (B1,issued,Issued+1,B2) ,
  set_value (B1,max,0,B3) )
],
[ (issuer="...") , (issued=0) , (max=40,000) ,
(keyholder="...") ,
(trustedissuer="...") ,
(digest="qZk+NkcGgWq6PiVxeFDcbJzQ2J0=") ,
(skey="D04951E4-332C-4693-B7DB-D3D1D1C20844") ,
(notbefore="2001-05-25T00:00:00") ,
(notafter="2002-05-25T00:00:00") ,
] )
] )

```

### LicenseScript rules:

```

play (Music,User) :
  license (Music,C,B1) -> license (Music,C,B2)
<= C |- canplay (B1,B2,User)
issue (Music,Issuer) :
  license (Music,C,B1) ->
  license (Music,C,B2) , license (Music,C,B3)
<= C |- canissue (B1,B2,B3,Issuer)

```

## B.17 Personal Copies

### XrML code:

```

-<license>
-<grant>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQAQAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<cx:copy/>
-<cx:digitalWork>
<cx:description xml:lang="en">
A very good book
</cx:description>
<cx:description xml:lang="fr">
Un tres bon livre
</cx:description>
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>

```

```

<cx:title>A Book of James</cx:title>
<cx:creator>James the first</cx:creator>
<cx:owner>Mike the man</cx:owner>
<cx:copyright>
Copyright 1999 Harper Collins Publishers.
All Rights Reserved.</cx:copyright>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
<cx:nonSecureIndirect URI=
"http://www.contentguard.com/bookOfJames.spd"/>
</cx:locator>
</cx:digitalWork>
-<allConditions>
-<sx:exerciseLimit>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>
E55129C1-74DF-40d0-B2A6-367AAB6AB05
</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:exerciseLimit>
-<validityInterval>
<notBefore>2001-05-25T00:00:00</notBefore>
<notAfter>2003-05-25T00:00:00</notAfter>
</validityInterval>
</allConditions>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCCo941tTExbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
alIDoedpL...
</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
-<details>
<timeOfIssue>
2000-01-27T15:30:00
</timeOfIssue>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (
http://www.contentguard.com/bookOfJames.spd,
[ (cancopy (B1,B2,B3,U) :-
  get_value (B1,keyholder,KH) ,
  get_value (B1,skey,SKey) ,
  seek_approve (U,KH,SKey) ,
  get_value (B1,notbefore,NB) ,

```

```

    get_value (B1, notafter, NA),
    today (D), D>NB, D<NA,
    get_value (B1, maxcopies, Max),
    get_value (B1, copied, Copied),
    Copied<=Max,
    set_value (B1, copied, Copied+1, B2),
    set_value (B1, maxcopies, 0, B3))
],
[ (keyholder="..."),
  (notbefore="2001-05-25T00:00:00"),
  (notafter="2003-05-25T00:00:00"),
  (maxcopies=3), (copied=0),
  (skey="E55129C1-74DF-40d0-B2A6-367AAB6AB05")
])

```

### LicenseScript rules:

```

copy (Ebook, User) :
  license (Ebook, C, B1) ->
  license (Ebook, C, B2), license (Ebook, C, B3)
<= C |- cancel (B1, B2, B3, User)

```

## B.18 Nested Revenue Model

### XrML code:

```

-<license>
-<inventory>
-<cx:digitalWork licensePartId="book1">
<cx:description xml:lang="en">
A very good book</cx:description>
<cx:description xml:lang="fr">
Un tres bon livre</cx:description>
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>A Book of James</cx:title>
<cx:creator>James the first</cx:creator>
<cx:copyright>
Copyright 1999 Harper Collins Publishers.
All Rights Reserved.</cx:copyright>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
<cx:nonSecureIndirect URI=
"http://www.contentguard.com/bookOfJames.spd"/>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grant>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>Efgao6NYf...</dsig:Modulus>
<dsig:Exponent>AQAQAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<cx:play/>
<cx:digitalWork licensePartIdRef="book1"/>
-<allConditions>
-<sx:fee>
-<sx:markup>
<sx:rate>0.10</sx:rate>
-<sx:fee licensePartId="baseFee">
-<sx:paymentPerUse>
<sx:rate currency="USD">5.00</sx:rate>
</sx:paymentPerUse>
-<sx:to>
-<sx:aba>
<sx:institution>139371581</sx:institution>
<sx:account>111111</sx:account>
</sx:aba>
</sx:to>
</sx:fee>

```

```

</sx:markup>
-<sx:to>
-<sx:aba>
<sx:institution>222371864</sx:institution>
<sx:account>123457</sx:account>
</sx:aba>
</sx:to>
</sx:fee>
</allConditions>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCo94ltTExbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
alIDoedpL...</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
-<details>
<timeOfIssue>2000-01-27T15:30:00</timeOfIssue>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (
http://www.contentguard.com/bookOfJames.spd,
[ (canplay (B1, B2, U) :-
  get_value (B1, keyholder, User), U=User,
  get_value (B1, paid, Paid), Paid=true,
  set_value (B1, paid, false, B2)),
  (canpay (B1, B2, KH, I1, A1, I2, A2, R, FR) :-
  get_value (B1, keyholder, KH),
  get_value (B1, institution1, I1),
  get_value (B1, institution2, I2),
  get_value (B1, account1, A1),
  get_value (B1, account2, A2),
  get_value (B1, rate, R),
  get_value (B1, fraction, FR),
  set_value (B1, paid, true, B2))
]),
[ (keyholder="..."), (paid=false),
  (rate=5.00), (fraction=0.10),
  (institution1="139371581"),
  (account1="111111"),
  (institution2="222371864"),
  (account2="123457"),
  (issuer="...")
])
wallet (
[ (canload (Bw1, Bw2, A) :-
  get_value (Bw1, money, Money),
  N is A+Money,
  set_value (Bw1, money, N, Bw2)),
  (canttransfer (Bw1, Bw2, KH, I1, A1, I2, A2, R, FR) :-
  get_value (Bw1, money, Money),

```

```

Money >= R,
transfer (R*FR, I1, A1),
transfer (R*(1-FR), I2, A2),
N is Money-R,
set_value (Bw1, money, N, Bw2) ],
[ (money=x) ]

```

### LicenseScript rules:

```

play (Movie, User) :
  license (Movie, C, B1) -> license (Movie, C, B2)
  <= C |- canplay (B1, B2, User)
pay (Ebook) :
  license (Ebook, C, B1), wallet (Cw, Bw1) ->
  license (Ebook, C, B2), wallet (Cw, Bw2)
  <= C |- canpay (B1, B2, KH, I1, A1, I2, A2, R, FR),
  Cw |- canttransfer (Bw1, Bw2, KH, I1, A1, I2, A2, R, FR)

```

## B.19 Accessing Web Service

### XrML code:

```

<-license>
<-grant>
<-keyHolder>
<-info>
<-dsig:KeyValue>
<-dsig:RSAKeyValue>
<dsig:Modulus>
cR0QWMWaq...</dsig:Modulus>
<dsig:Exponent>
AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<po:transmitDocuments/>
<-serviceReference>
<-uddi>
<-serviceKey>
<uuid>
EE4A90A5-8AC9-4f31-85F7-6619AA573449
</uuid>
</serviceKey>
</uddi>
</serviceReference>
<-allConditions>
<-po:permittedDocumentType>
<-po:schema>
<po:namespace>
http://www.xrml.org/schema/2001/11/po
</po:namespace>
<po:type>PURCHASE_ORDER</po:type>
</po:schema>
</po:permittedDocumentType>
<-po:documentLimitation>
<-po:guard>
<-po:schema>
<po:namespace>
http://www.xrml.org/schema/2001/11/po
</po:namespace>
<po:type>PURCHASE_ORDER</po:type>
</po:schema>
</po:guard>
<-po:limitations>
<po:pattern>
purchaseOrder/Total < 1000
</po:pattern>
<-po:credentialRevocationCheck>
<po:pattern>purchaseOrder/Total
>= 25
</po:pattern>
<-revocationFreshness>
<maxIntervalSinceLastCheck>
P1D
</maxIntervalSinceLastCheck>
</revocationFreshness>
</po:credentialRevocationCheck>

```

```

<-po:credentialRevocationCheck>
<po:pattern>
purchaseOrder/Total < 25
</po:pattern>
<-revocationFreshness>
<noCheckNecessary/>
</revocationFreshness>
</po:credentialRevocationCheck>
</po:limitations>
</po:documentLimitation>
<-validityInterval>
<notAfter>2003-05-24T16:20:00</notAfter>
</validityInterval>
</allConditions>
</grant>
<-issuer>
<-dsig:Signature>
<-dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<-dsig:Reference>
<-dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCCo941tTExbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
AYmqOhSHb...
</dsig:SignatureValue>
<-dsig:KeyInfo>
<-dsig:KeyValue>
<-dsig:RSAKeyValue>
<dsig:Modulus>
X0j9q90Ax...</dsig:Modulus>
<dsig:Exponent>
AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
<-details>
<timeOfIssue>
2001-05-24T16:20:00</timeOfIssue>
<-revocationMechanism>
<-querySignature>
<-uddi>
<-serviceKey>
<uuid>
1F2CBD46-CE56-4422-A2D5-F5E04DF94C6A
</uuid>
</serviceKey>
</uddi>
</querySignature>
</revocationMechanism>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (purchases_order,
[ (canrevoke (B1, B2) :-
  get_value (B1, keyholder, KH),
  get_value (B1, issuer, I),
  get_value (B1, issueddate, I_d),
  get_value (B1, notafter, NA),
  today (D), I_d<D+1, D<=NA,
  get_value (B1, purchaseorder, PO), PO<1000),
  (canrevoke (B1, B2) :-
  get_value (B1, keyholder, KH),
  get_value (B1, issuer, I),
  get_value (B1, issueddate, I_d),

```

```

    get_value(B1,notafter,NA),
    today(D), I_d<D+1, D<=NA,
    get_value(B1,purchaseorder,PO),
    PO>=25, credentialRevokeCheck(PO,KH,I)),
  (canrevoke(B1,B2) :-
    get_value(B1,keyholder,KH),
    get_value(B1,issuer,I),
    get_value(B1,issueddate,I_d),
    get_value(B1,notafter,NA),
    today(D), I_d<D+1, D<=NA,
    get_value(B1,purchaseorder,PO),
    PO<25, credentialRevokeCheck(PO,KH,I)),
  ],
  [
    (keyholder="..."), (issuer="..."),
    (purchaseorder=nil),
    (issueddate="2001-05-24T16:20:00"),
    (notafter="2003-05-24T16:20:00"),
    (skey="1F2CBD46-CE56-4422-A2D5-F5E04DF94C6A")
  ]
)

```

### LicenseScript rules:

```

revoke(POrder) :
  license(POrder,C,B1) -> license(POrder,C,B2)
  <= canrevoke(B1,B2)

```

## B.20 Software Execution

### XrML code:

```

-<license>
-<inventory>
-<cx:digitalWork licensePartId="pico">
-<cx:locator>
-<secureIndirect URI=
"http://foo.com.mainframe/pico.exe">
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
0kccZ4a3zFW9OPT1qEIqSg==
</dsig:DigestValue>
</secureIndirect>
</cx:locator>
</cx:digitalWork>
-<keyHolder licensePartId="fooMainframe">
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>n5gzmvv4/...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
</inventory>
-<grant>
-<keyHolder licensePartId="Alice">
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>n5gzmvv4/...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<cx:execute/>
<cx:digitalWork licensePartIdRef="pico"/>
-<cx:source>
-<keyHolder licensePartIdRef="fooMainframe">
</keyHolder>
</cx:source>
</grant>
</license>

```

### LicenseScript code:

```

license(http://foo.com.mainframe/pico.exe,
  [
    (canexecute(B1,B2,U) :-
      get_value(B1,mainframe_kh,Main),
      identify(M), M=Main,
      get_value(B1,user_kh,User),
      U=User)
  ],
  [
    (digest="0kccZ4a3zFW9OPT1qEIqSg=="),
    (mainframe_kh="foo..."),
    (user_kh="alice...")
  ])

```

### LicenseScript rules:

```

execute(SWare,User) :
  license(SWare,C,B1) -> license(SWare,C,B2)
  <= C |- canexecute(B1,B2,User)

```

## B.21 Confidentiality of Rights

### XrML code:

```

-<license>
-<grant>
<cx:play/>
-<cx:digitalWork licensePartId="dvdMovie">
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Air Force One</cx:title>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
-<secureIndirect URI=
"http://sonyPictures.com/AirForceOne">
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
0kccZ4a3zFW9OPT1qEIqSg==</dsig:DigestValue>
</secureIndirect>
</cx:locator>
</cx:digitalWork>
-<sx:fee>
-<sx:paymentFlat>
<sx:rate currency="USD">10.00</sx:rate>
-<sx:paymentRecord>
-<sx:stateReference>
-<uddi>
-<serviceKey>
<uuid>D04951E4-332C-4693-B7DB-D3D1D1C20844</uuid>
</serviceKey>
</uddi>
</sx:stateReference>
</sx:paymentRecord>
</sx:paymentFlat>
</sx:fee>
</grant>
-<grant>
-<encryptedGrant Type=
"http://www.w3.org/2001/04/xmenc#Content">
<enc:EncryptionMethod Algorithm=
"http://www.w3.org/2001/04/xmenc#rsa-1_5"/>
-<dsig:KeyInfo>
<dsig:KeyName>Alice</dsig:KeyName>
</dsig:KeyInfo>
-<enc:CipherData>
<enc:CipherReference URI="cid:33"/>
</enc:CipherData>
</encryptedGrant>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

```

```

-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCo941tTExbj1/Q==</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
alIDoedpL...</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
-<details>
<timeOfIssue>2000-01-27T15:30:00</timeOfIssue>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (http://sonyPictures.com/AirForceOne,
[ (canplay (B1, B2) :-
  get_value (B1, paid, Paid),
  Paid=true,
  get_value (B1, played, Played),
  get_value (B1, maxplays, Max),
  Played<Max,
  set_value (B1, played, Played+1, B2)),
 (canpay (B1, B2, I, R) :-
  get_value (B1, rate, R),
  get_value (B1, issuer, I),
  set_value (B1, paid, true, B2)),
 (candecrypt (B1, B2) :-
  get_value (B1, userkey, Ukey),
  get_value (B1, enc_right_ref, Ref),
  decrypt (Ref, Ukey, Rights))
],
[ (rate=10.00), (paid=false),
(issuer="..."),
(played=0), (maxplays=1),
(skey="D04951E4-332C-4693-B7DB-D3D1D1C20844"),
(userkey="alice..."),
(enc_right_ref="cid:33"),
(digest="PB4QbKQCo941tTExbj1/Q=="),
(signature="alIDoedpL...")
(sigkey="<mod>g8NRYMG30...<exp>AQABAA=="),
])
wallet (
[ (canload (Bw1, Bw2, A) :-
  get_value (Bw1, money, Money),
  N is A+Money,
  set_value (Bw1, money, N, Bw2)),
 (cantransfer (Bw1, Bw2, R, KH, Skey, I) :-
  get_value (Bw1, money, Money),
  Money >= R,
  transfer (R, I),
  trackrecord (Skey, I, KH),
  N is Money-R,
  set_value (Bw1, money, N, Bw2)) ],
[ (money=x) ])

```

### LicenseScript rules:

```

play (Movie) :
  license (Movie, C, B1) -> license (Movie, C, B2)
<= C |- canplay (B1, B2)
decrypt (Movie) :
  license (Movie, C, B1) -> license (Movie, C, B2)

```

```

<= C |- candecrypt (B1, B2)
pay (Movie) :
  license (Movie, C, B1), wallet (Cw, Bw1) ->
  license (Movie, C, B2), wallet (Cw, Bw2)
<= C |- canpay (B1, B2, I, R),
  Cw |- cantransfer (Bw1, Bw2, I, R)

```

## B.22 Watermark and Content Fragments

### XrML code:

```

-<license>
-<inventory>
-<cx:digitalWork licensePartId="logo">
<cx:description>watermark logo</cx:description>
-<cx:locator>
-<secureIndirect URI=
"http://www.usc.edu/trojan.bmp">
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PlccW6bazFW9OPT1qEIqSk==</dsig:DigestValue>
</secureIndirect>
</cx:locator>
</cx:digitalWork>
</inventory>
-<grant>
-<keyHolder>
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus> p8sN4Kee...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<cx:print/>
-<cx:digitalWork>
<cx:description>
Alice In the Wonder Land</cx:description>
</cx:digitalWork>
-<allConditions>
-<school:content>
<school:unit type="onix:NumberOfPages"/>
<school:from>1</school:from>
<school:to>10</school:to>
</school:content>
-<cx:watermark>
<cx:user-name/>
<cx:string>
Title: 'Alice In the Wonder Land' </cx:string>
<cx:render-location/>
<cx:render-time/>
<cx:object licensePartIdRef="logo"/>
</cx:watermark>
</allConditions>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
-<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xrml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCo941tTExbj1/Q==</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>alIDoedpL...</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>

```

```

-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
<-details>
<timeOfIssue>2000-01-27T15:30:00</timeOfIssue>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (http://www.usc.edu/trojan.bmp,
[ (canprint (B1,B2,P1,P2) :-
  get_value (B1,page_fr,Pf),
  get_value (B1,page_to,Pt),
  P1>=Pf, P2<=Pt,
  get_value (B1,watermark_obj,Wobj),
  get_value (B1,watermark_str,Wstr),
  verify_watermark (Wobj,Wstr)
],
[ (keyholder="..."), (issuer="..."),
  (watermark_str="Title: 'Alice In the Wonder Land'"),
  (watermark_obj="logo"),
  (digest="PlccW6bazFW9OPT1qEIqSk=="),
  (page_fr=1), (page_to=10)
])

```

### LicenseScript rules:

```

print (Bmp,P1,P2) :
  license (Bmp,C,B1) -> license (Bmp,C,B2)
<= C |- canprint (B1,B2,P1,P2)

```

## B.23 Secure Device

### XrML code:

```

-<license>
-<inventory>
-<keyHolder licensePartId="trustedIssuer">
-<info>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
</inventory>
-<grant>
-<forall varName="projectorEpson">
-<everyone>
<school:projector/>
-<trustedIssuer>
<principal licensePartIdRef="trustedIssuer"/>
</trustedIssuer>
</everyone>
-<everyone>
-<school:kernelSecurityLevel>
<cx:value>5</cx:value>
</school:kernelSecurityLevel>
-<trustedIssuer>
<principal licensePartIdRef="trustedIssuer"/>
</trustedIssuer>
</everyone>
</forall>
-<forall varName="studentGroup">
-<everyone>
<sx:dnsName>www.usc.edu</sx:dnsName>
-<trustedIssuer>
<principal licensePartIdRef="trustedIssuer"/>
</trustedIssuer>

```

```

</everyone>
</forall>
<cx:play/>
-<cx:digitalWork>
-<cx:metadata>
-<xml>
-<cx:simpleDigitalWorkMetadata>
<cx:title>Alice in
the Wonder Land</cx:title>
<cx:copyright>Copyright 1999 Addison Wesley.
All Rights Reserved.</cx:copyright>
<onix:mediaFileTypeCode>
movie0001
</onix:mediaFileTypeCode>
</cx:simpleDigitalWorkMetadata>
</xml>
</cx:metadata>
-<cx:locator>
-<secureIndirect URI=
"http://sonyPictures.com/AliceInTheWonderLand">
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
0kccZ4a3zFW9OPT1qEIqSg==
</dsig:DigestValue>
</secureIndirect>
</cx:digitalWork>
-<cx:renderer>
-<keyHolder varRef="projectorEpson">
</keyHolder>
</cx:renderer>
</grant>
-<issuer>
-<dsig:Signature>
-<dsig:SignedInfo>
<dsig:CanonicalizationMethod Algorithm=
"http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<dsig:SignatureMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<dsig:Reference>
-<dsig:Transforms>
<dsig:Transform Algorithm=
"http://www.xml.org/schema/2001/11/xrml2core#license"/>
</dsig:Transforms>
<dsig:DigestMethod Algorithm=
"http://www.w3.org/2000/09/xmldsig#sha1"/>
<dsig:DigestValue>
PB4QbKQCo941tTExbj1/Q==
</dsig:DigestValue>
</dsig:Reference>
</dsig:SignedInfo>
<dsig:SignatureValue>
zIRYaxl5E...
</dsig:SignatureValue>
-<dsig:KeyInfo>
-<dsig:KeyValue>
-<dsig:RSAKeyValue>
<dsig:Modulus>g8NRYMG30...</dsig:Modulus>
<dsig:Exponent>AQABAA==</dsig:Exponent>
</dsig:RSAKeyValue>
</dsig:KeyValue>
</dsig:KeyInfo>
</dsig:Signature>
<-details>
<timeOfIssue>2001-01-27T15:30:00</timeOfIssue>
<-validityInterval>
<notBefore>2000-02-03T17:26:00</notBefore>
<notAfter>3000-02-03T17:26:00</notAfter>
</validityInterval>
</details>
</issuer>
</license>

```

### LicenseScript code:

```

license (http://sonyPictures.com/AliceInTheWonderLand,
[ (canplay (B1,B2,User) :-
  get_value (B1,notbefore,NB),
  get_value (B1,notafter,NA),

```

```
        today(D), D>NB, D<NA,
        get_value(B1,renderer,R),
        identify(Renderer), R=Renderer,
        get_value(B1,studentgroup,SG),
        member(User,SG)
    ],
    [
        (trustedissuer="..."),
        (issuer="..."),
        (notbefore="2000-02-03T17:26:00"),
        (notafter="3000-02-03T17:26:00"),
        (renderer=projectorEpson_kh),
        (digest="0kccZ4a3zFW9OPT1qEIqSg=="),
        (studentgroup=[...])
    ])
}
```

### LicenseScript rules:

```
play(Movie,User) :
    license(Movie,C,B1) -> license(Movie,C,B2)
<= canplay(B1,B2,User)
```