DRM Use License Negotiation using ODRL v2.0

Alapan Arnab, Andrew Hutchison
Data Network Architectures Group
Department of Computer Science
University of Cape Town
Rondebosch, 7701
South Africa

{aarnab,hutch}@cs.uct.ac.za

Abstract

In [9], Camp discussed why DRM is not equivalent to copyright enforcement. In 2005, Arnab et al. discussed how DRM is in fact the enforcement of licensing agreements, and promoted the use of negotiation in DRM as a mechanism to handle fair use scenarios [3].

In this paper, we detail negotiation protocols for two of the three types of negotiation – bidding and bargaining (the third type, auctioning, can easily be handled without any new technology). We motivate the correctness and completeness of our protocols through the use of Petri net modeling. We also motivate the use of the latest draft of the ODRL v2.0 rights expression language (REL) as a language for expressing negotiations in DRM systems. By using a REL in the protocol specifications we remove the need to translate between the protocol and the rights expression language, thus speeding up the overall license acquisition process and reducing the risk of translation errors.

1 Introduction

Rights Expression Languages (RELs) are often referred to as the most crucial component of DRM systems [16]. For this reason, there has been a lot of focus on developing RELs. In [11], Coyle distinguished RELs into three different categories:

1. expression of *copyright*
2. expression of *contract* or *license agreements*
3. control over *access* and/or *use*

Camp argued in [9] that DRM does not provide copyright management, and is not a mechanism to control copyright. In [3], the authors argue that DRM should not be seen as a mechanism to enforce copyright law but rather as a mechanism to enforce contracts on access and usage of digital data. In such a view, the primary role of a REL is not to express copyright but rather to express contractual agreements. Consequently, DRM systems can then be seen as the enforcement of such contracts. In [11], Coyle concluded that none of the current RELs – including general purpose RELs such as ODRL and MPEG-21/5 – have the full functionalities required for such purposes. For example, the lack of bi-directional expression (from the user to the rights holder) has been cited as a deficiency in RELs by Mulligan et al. in [18].

The lack of bi-directional expression in RELs has meant that there is no formal mechanism for the user to communicate to the rights holders. This has meant, that one of the crucial parts of the contractual process – negotiation – is not possible. In [2], Arnab et al. proposed bi-directional extensions to ODRL 1.1, and they had previously discussed similar extensions for XrML in [1]. Negotiations were one of their proposed mechanisms for a technical solution to "*fair use*" in [3] where they also discussed a basic protocol to conduct negotiations.

We believe that negotiation of use licenses is closely linked with representation of the license, and thus Rights Expressions Languages (RELs) have a crucial role in this regard. This removes the need to convert between expression of negotiation terms and the final agreement, thus reducing errors that are possibly introduced during translation. However, negotiation capabilities for the REL should minimise the complexity should negotiations not be utilised (as in most current DRM systems).

In [20], Pruitt defines negotiation as "*a form of decision making in which two or more parties talk with one another in an effort to resolve their opposing interests*". Sharrock defines a contract as "*an agreement which creates an obligation or obligations between the parties to the agreement*" [22]. Thus negotiation is an important component of the contractual process, and if DRM is seen as a contractual process, there is a need to support negotiation in DRM.

1.1 Scope and Contribution

In [24] Su et al. also discussed two important components for electronic negotiations:

- a formal protocol, and
- an effective agent to carry out the task ([24] concentrated on AI agent strategies for this task).

In [5], Bartolini et al. added two further requirements for automated negotiation:

- a language to define rules of negotiation which can be used by agents to evaluate negotiation

proposals, and

- a language to express negotiation proposals.

In this paper, we introduce two different protocols to conduct negotiations, which uses the latest draft of the ODRL v2.0 specification to represent negotiation proposals. Both of these protocols are more complex than the protocol discussed in [3]. Furthermore, we use Coloured Petri Nets to prove the integrity of the protocols, by proving that there are no deadlocks and that all the desired states of the protocol are reachable.

This paper does not discuss agent decision making strategies nor languages that can be used to define agent behaviour.

1.2 Organisation

In the next section we briefly discuss related work in negotiation protocols. We then discuss the various types of negotiations in section 3 followed by a discussion on the requirements for negotiation in DRM systems. Sections 5 and 6 discuss our two protocols, including the Petri net modelling results. Following this, we motivate the use of ODRL v2 as the language to represent negotiations before analysing our protocols with the requirements defined in section 4 and drawing our conclusions.

2 Related Work

There has been a great deal of research into negotiation protocols, tactics and other related fields. In economics and mathematics, game theory models have been used to discuss different tactics that could be used to arrive at the most rational outcome [17, 8]. In the social sciences, there has been a lot of research in how different parties act during negotiations and how these actions affect the eventual outcome [21, 20]. In Computer Science, the bulk of research in negotiation has focused on agent negotiation, focusing on agent decision tactics, efficiency and protocols [17]. Some of this research has been extended to e-commerce scenarios [24].

Negotiation protocols have also been used in other aspects of electronic communication. For example, the Secure Socket Layer (SSL) protocol has a negotiation component in its handshake protocol to set-up encryption and MAC algorithms [23]. However, in most of these cases, the number of negotiable factors are small, and there are often strict guidelines which dictate the result of the negotiation.

In [21], Raiffa discusses a number of factors that affect negotiation strategy and protocol. These factors include:

1. The number of parties,
2. parties negotiating on behalf of a group,
3. repetitiveness of the negotiation process and its effect on reputation,
4. the number of terms being negotiated,
5. 3^{rd} party involvement

The use of negotiation in consumer or end-user environment is almost non-existent. In [13], Elfatatry et al. suggests the use of negotiation to tailor software products in a Web Services environment, but does not specify any protocols for such a service. This paper details the use of negotiations in an end-user environment, and this poses a further challenge, not catered for in most negotiation scenarios – the protocol must be able to cater for three different types of interactions:

1. The human end user and an agent representing the license holder
2. The human end user and a human representing the license holder
3. An agent representing the end user and an agent representing the license holder

Most negotiation protocols are developed for agents, and they are often moulded specifically for the agent's purpose. Furthermore, these protocols often incorporate the agent's decision making processes. For these reasons, existing negotiation protocols developed for agents are not fully appropriate for the scenario we present in this paper. The negotiation protocol must however take into account all the factors discussed by Raiffa, and the protocols we present in this paper are similar to argumentation based models described by Jennings et al. in [17] and the language syntax has similar properties to the logic based language detailed in [26, 25].

An important criteria for any protocol design, is to ensure the integrity and correctness of the protocol. Petri nets have been widely used to model and formally represent discrete distributed systems [7], and we have chosen to use coloured Petri nets (which are a subset of Petri nets) to model our protocols. Petri nets are place-transition nets comprising of a non-empty set of places, transitions, arcs connecting places to transitions and tokens to define the value of a given place. Coloured Petri nets provide for systems with more than one type for a given place, and any coloured Petri net can be described as a traditional Petri net [7].

In [7], the authors discuss the following properties of Petri nets, and for each property, we discuss its implications in modelling a communications protocol.

1. **Reachability:** The reachability set of a Petri net is the set of all possible states achievable for a given system. Proving that a Petri net is reachable implies that every state in the

associated system is achievable. Furthermore, reachable Petri nets are required before a system can have steady state distribution [7] (where the performance of the system is not affected regardless of the number of iterations of the protocol run, e.g. lack of buffer overflows). For these reasons, it is highly desirable to have reachability in communication protocols.

2. **Liveness:** A Petri net is live if there is at least one possible transition between two different states of the net. A protocol whose Petri net is not live, has a deadlock and the protocol cannot continue to execute. Thus, it is necessary for a protocol to have a live Petri net.

3. **Boundedness:** A Petri net is bounded if there is a maximum number of tokens, regardless of the number of iterations. Thus, an unbounded net often indicates a flaw in protocol, like the possibility of buffer overflows. Thus, it is necessary for a protocol to have a bound Petri net.

4. **Safety:** A Petri net is safe if the maximum number of tokens is 1. While it is nice to have safe nets, it is not necessary to have a safe net for a protocol.

Circular deadlock detection is not a direct property of Petri nets. However, if a Petri net representation of a protocol is *bound* and *reachable*; then it follows that the protocol does not have a circular deadlock. This follows from the fact that, if there is a circular deadlock, certain states of the Petri net will no longer be reachable.

Petri net modelling for our protocols cannot prove that they are correct in their intended functions, but can prove that they have no obvious flaws. It is for this purpose, that we have used Petri net modelling. We have used a well known Petri net tool, CPNTools[1], version 2.0.0 for GNU-Linux, to create and analyse our Petri nets.

3 Types of Negotiations

As discussed earlier, negotiation can be defined as a process whereby a contract is concluded. In [24], the authors distinguish three types of negotiations:

1. **Bidding:** The buyer specifies the service or product that he needs and asks bids from potential suppliers. The buyer then selects one or more of the suppliers to provide the service or product. Currently, no DRM system can support bidding.

2. **Auction:** The auction can be viewed as the opposite of bidding where the supplier of the product or service promises to perform the service or deliver the goods to the customer

[1] http://wiki.daimi.au.dk/cpntools/cpntools.wiki

with the highest bid. There are a variety of auction types, and current DRM systems should be able to handle auctions as price is the only "negotiable" component of an auction. For this reason, we do *not* discuss auctions any further in this paper.

3. **Bargaining:** Bargaining is the most flexible type of negotiation allowing all the parties involved to dynamically change the terms and conditions to suit their needs. A lot of research in negotiations has focused in this area.

Current DRM systems only support transactions where the suppliers determine a fixed price for the product under fixed terms and conditions. There is no scope for bargaining. While these types of transactions are largely fine for most consumer oriented digital data (for example music), they are not useful for automating business use of digital data or for more non-consumer oriented use of digital data (for example large volume purchases for academic usage). As discussed in [3], bargaining could also be used as a mechanism to assure fairer usage of digital media, and opens up possibilities for allowing "fair use" scenarios not possible with current DRM systems.

4 Requirement for Negotiation for DRM

As discussed in section 2, Raiffa detailed a number of factors that need to be considered when determining strategies and protocols for DRM. In this section, we look at how these factors apply to DRM systems, and also discuss the requirements specified by the Digital Media Project (DMP)[2] in [12]. The DMP is a collaborative project between various interested parties, including device manufacturers and software vendors, to create a standardised platform for interoperable media DRM systems [10].

4.1 Factors affecting Negotiation in DRM

4.1.1 The number of parties

As identified by Bartolini et al. [6] and also by the DMP [10], there are a number of parties involved in the DRM value chain. The DRM value chain, as defined by the DMP is shown in figure 1. The negotiation protocols we present are focused for *two* parties – one party that has the right and ability to conclude use license agreements (usually the rights holder) and the consumer of the license (which could also be the producer or service producer).

[2] http://www.dmpf.org/

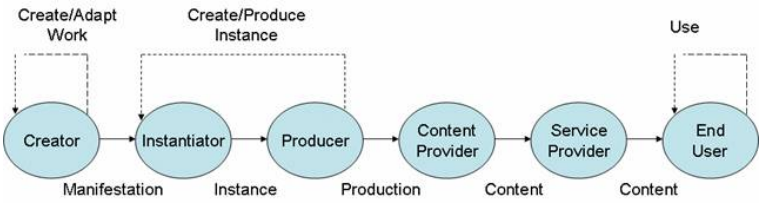


Figure 1: The DRM Value Chain [10]

4.1.2 Parties negotiating on behalf of a group

Although only two parties are negotiating the use license, both parties could be representing larger groups. For example, in consumer music, a parent could be acquiring music that will be accessed by all the members of his/her immediate family.

4.1.3 Repetitiveness of the negotiation process and its effect on reputation

This factor is more relevant to the strategy employed by the parties and not to the protocols themselves, and thus not discussed in detail in this paper. However, it must be noted that using reputation of the parties as part of the negotiation process could create different business strategies, not currently pursued. For example, users associated to well known organisations could be seen as more reputable, and thus given more rights than lesser known users. In [16], the authors proposed the use of reputation to determine the level of security for DRM packages.

4.1.4 The number of issues

Pruitt defines an issue as the topic under discussion [20]. Thus, in a DRM license, each permission or right is a separate issue, unless they are bundled together (e.g. the licensee can get read and play rights together or not at all). Individual terms of a specific right (e.g. restrictions on the number of times that right can be exercised) is the subject of negotiation, and not separate issues. Issues are often linked together, and an issue can often influence the negotiation position of a party for another issue (for example, the number of users covered by the license can affect the price of the license). Because, the number of possible rights in a DRM use license, and other issues like validity of the license and the number of users covered by the license, DRM use license negotiation can be very complex.

4.1.5 3^{rd} party involvement

In [3], Arnab et al. proposed the use of copyright tribunals to arbitrate use license disputes, and this could be a possible 3^{rd} party involved in the negotiations. In a subsequent paper [4], the authors noted that users are more willing to use negotiation as a mechanism to enable fair use if the process is monitored by independent 3^{rd} parties.

4.2 DMP requirement for negotiations

In [12], the DMP listed the following requirements for negotiations in DRM systems:

1. End-users can express their agreement or disagreements with proposed license terms.
2. The protocol shall support changes to any parameter of the license.
3. The protocol shall support automatic negotiation of license terms.
4. At every step a human readable license must be provided.
5. The protocol shall enable the setting of certain parameters as non subject of negotiation.
6. The protocol shall allow the determination of the degree of confidentiality (no eavesdrop) of the protocol.
7. The protocol shall not require revealing the real identities until the protocol has been successfully concluded.

5 Bidding

Bidding does not have much impact for consumer oriented DRM products, but could have massive impact in business transactions conducted over the Internet. For example, an advertising agency could be looking for classical music to accompany their television advertisements. For this purpose, they create a tender inviting musicians, bands etc. to supply the music under certain terms. Prospective parties can then formulate their offers, possibly offering different terms (for example a larger catalogue of music) and their prices. The advertising agency can then consider the offers and make their choice accordingly. This type of scenario cannot be handled by current DRM systems, nor by the simple negotiation protocol proposed in [3].

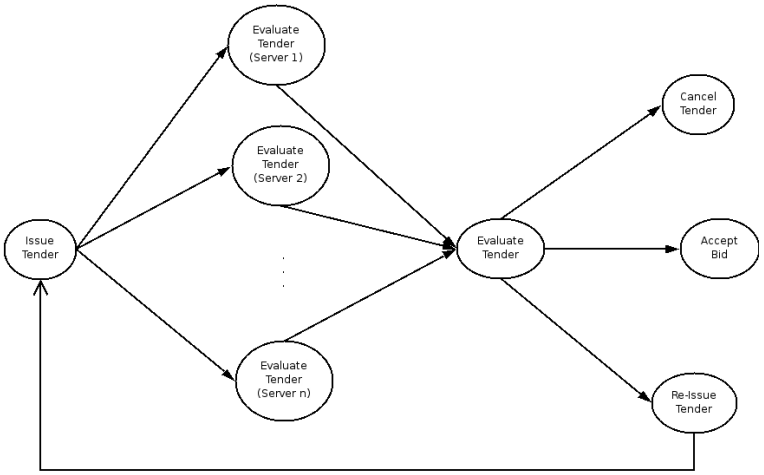


Figure 2: Bidding Process

5.1 Process

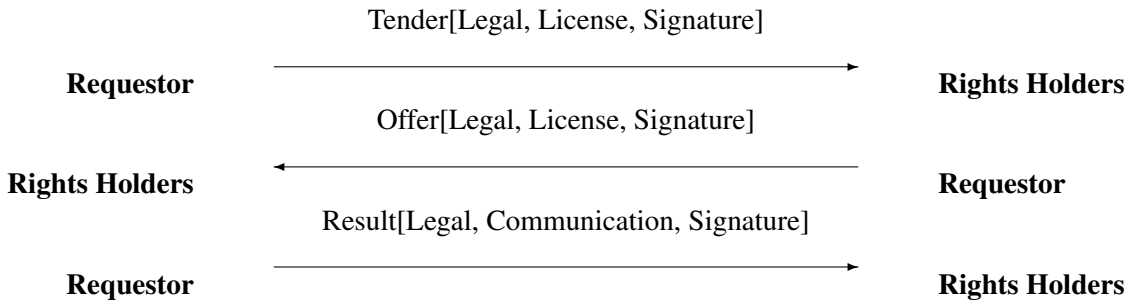
A flowchart showing the bidding process is presented in figure 2. The client issues a *tender* outlining the data they are interested in, the terms and conditions they would like for having access to such data and the time limit for responses to the set of possible respondents. The bidders then evaluate the tenders and create appropriate offers (if interested) outlining their price. The client can then evaluate the various offers and choose a winner, re-tender or close down the tender process and not choose any of the bids. Feedback is not mandatory, although it could be good business practice to outline why a tender is rejected.

5.2 Protocol

By its nature, bidding is not an interactive, instantaneous process. As shown in figure 2, there are only three parts to a bidding process:

1. The *announcement* of the tender requesting offers
2. The *submission* of offers
3. The *notification* of the outcome

For this reason, the bidding protocol has the following simple high level structure.



5.2.1 Message Format

The protocol is envisaged to run as SOAP messages between the two parties, with each step comprising of an XML encoded message between the parties. The exact representation of the message will be using ODRL v2.0 and this is discussed in more detail in section 7. The message can have a number of different components, and these are detailed below:

- **Communication:** This component comprises of the message elements that can be used to communicate between the parties, including the acceptance and rejection of offers.
- **Legal:** Use licenses are legal contracts, and there may be legal terms than need to be expressed, which do not form part of the main use license, for example liability disclaimers.
- **License:** The license forms the core component of the use license, and is effectively the terms and conditions being negotiated (like permission to play, read, modify etc).
- **Signature:** It is of paramount importance, that integrity of the communication is maintained. While, some communication protocols have integrity checking, this is not guaranteed. For this reason, we feel it is necessary to have a digital signature component for the message. Digital signatures also provide for non-repudiation, and are now considered legally binding in many countries. The entire message should be signed.

The same message content and notation is used in our bargaining protocol detailed in the next section.

5.3 Modelling

Figure 3 shows the Petri net model for our bidding protocol. Certain steps in the protocol can have a number of different outcomes, which we have modelled using the discrete function in CPNTools.

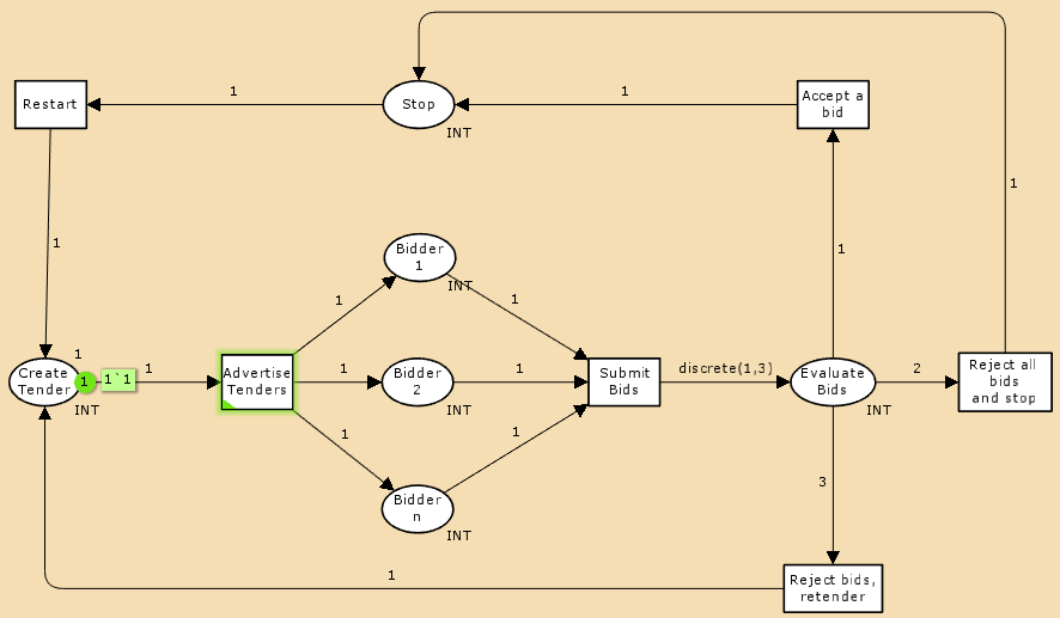


Figure 3: Bidding Petri Net

This function is a shortcut random number function available in CPNTools, and has been used to keep the net compact.

We have used the state space analysis component of CPNTools to analyse the net. Analysis shows us that the Petri net is live, reachable and safe. This means that every state of the protocol is reachable with no deadlocked states. Because the net is safe, there will be no performance degradation in repetitive iterations of the protocol, although that scenario should not arise in a bidding scenario.

6 Bargaining

Bargaining is the most complex negotiation strategy, but it is also the most powerful as it can be used not only for new use licenses but also to change existing use licenses. In [3], the authors discussed the use of bargaining as a mechanism to enable fairer use licenses for consumers, and as far as we are aware, this is the only technical approach to fair use for DRM systems.

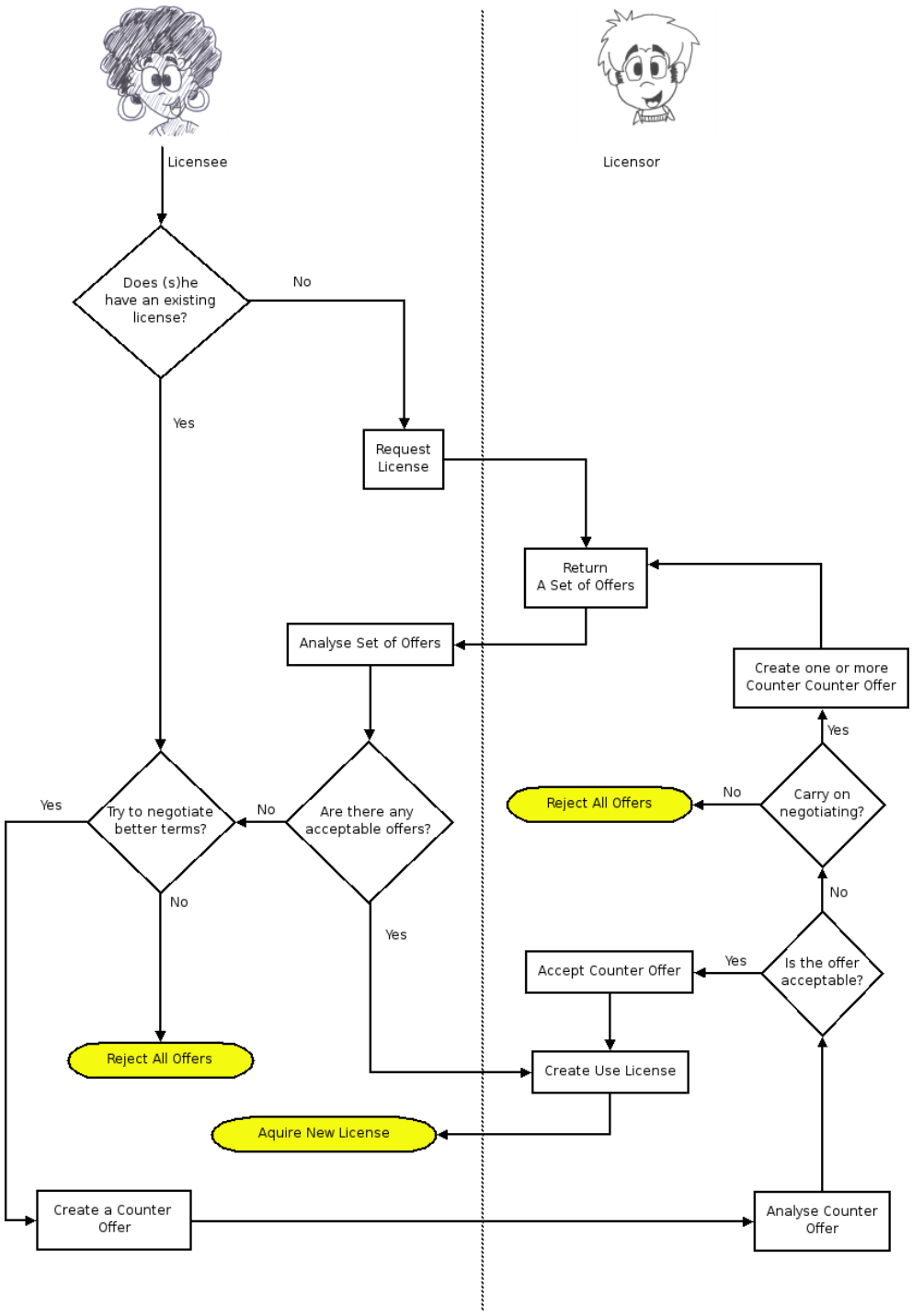


Figure 4: Flowchart for a bargaining protocol

6.1 Process

A bargaining protocol is shown in figure 4 and is a refinement of the simple negotiation protocol discussed in [2, 3] and is similar to the argumentation protocols discussed by Jennings et al. in [17]. The protocol assumes that the client communicates with the correct rights holder (or appropriate representative). It also assumes that the rights holder is initially willing to offer a license agreement to the client. Catering for the above two scenarios is not shown in the protocol but is trivial to handle.

Analysis of a request, or counter-offers, from clients should depend on the business scenario presented by the client. The protocol can handle anonymous clients, but anonymity may not be an ideal bargaining position for the client. For example, a well established client, with a long history of business association, could get more favourable terms and conditions compared to a new client. In such a case, the knowledge of the client's identity is required before the rights holder makes their decision. Similarly, a client wanting a transaction of high monetary value could be allowed a discount. In this case, anonymity of the client can be preserved unless there is a legal requirement forbidding anonymity of the client. As discussed earlier, the business logic used for negotiation is not discussed in this proposal, but needs to be addressed before the full power of bargaining is realised.

6.2 Protocol

As shown in figure 4, the bargaining protocol is more complicated than the bidding protocol, mainly due to the two different start and end possibilities. But, since one of the start possibilities is also part of the general bargaining protocol (does the client have an existing license), we need to show only the one path for the protocol.

1. The client requests for a new license. The identifier for the digital resource can be communicated using the "communication" element.

   Request[Legal, Communication, Signature]

   Requestor ⟶ **Rights Holder**

2. The rights holder sends back n offers (where n is a positive integer) to the requestor.

   n, Offer[Legal, License, Signature]

   Rights Holder ⟶ **Requestor**

3. After analysing the offers, the requestor can do one of the following:

(a) Accept one or more of the offers. Use of multiple licenses for the same digital resource is currently not handled by any DRM system, but there should be no reason why this should not be possible.

n, Acceptance[Legal, Communication, Signature]

Requestor ⟶ **Rights Holder**

(b) Reject all the offers, and quit negotiations.

Rejection[Legal, Communication, Signature]

Requestor ⟶ **Rights Holder**

(c) Reject all the offers, and enter negotiations, based on one of the offers or create counter offers from scratch. In the later case, a Counter-Offer is created instead of a Request. The requestor can create multiple requests or counter offers.

Rejection[Legal, Communication, Signature], n,
Offer[Legal, License, Signature]

Requestor ⟶ **Rights Holder**

4. Depending on the requestor's response, the rights holder does one of the following:

(a) If the requestor rejects all offers, the rights holder closes down the negotiation system.

(b) If the requestor proposes a counter offer, the rights holder can:

i. Reject all proposals and close down negotiations.

Rejection[Legal, Communication, Signature]

Rights Holder ⟶ **Requestor**

ii. Reject all proposals, but carry on with negotiations by creating offers that try to satisfy the counter offers.

Rejection[Legal, Communication, Signature], n,
Offer[Legal, License, Signature]

Rights Holder ⟶ **Requestor**

iii. Accept one or more of the proposals (see the next step).

(c) If the requestor accepted an offer, or the rights holder accepts one or more of the counter proposals:

n, Agreement[Legal, License, Signature]

Rights Holder ⟶ **Requestor**

All other offers are deemed to have been rejected.

5. Depending on the rights holder's actions:

   (a) If the rights holder wishes to close down negotiations, the requestor also closes down its negotiation system.

   (b) If the rights holder wishes to carry on with negotiations, the protocol goes back to step 1.

   (c) If the rights holder offers agreements, store the agreements and close down the negotiation system.

6.3 Modelling

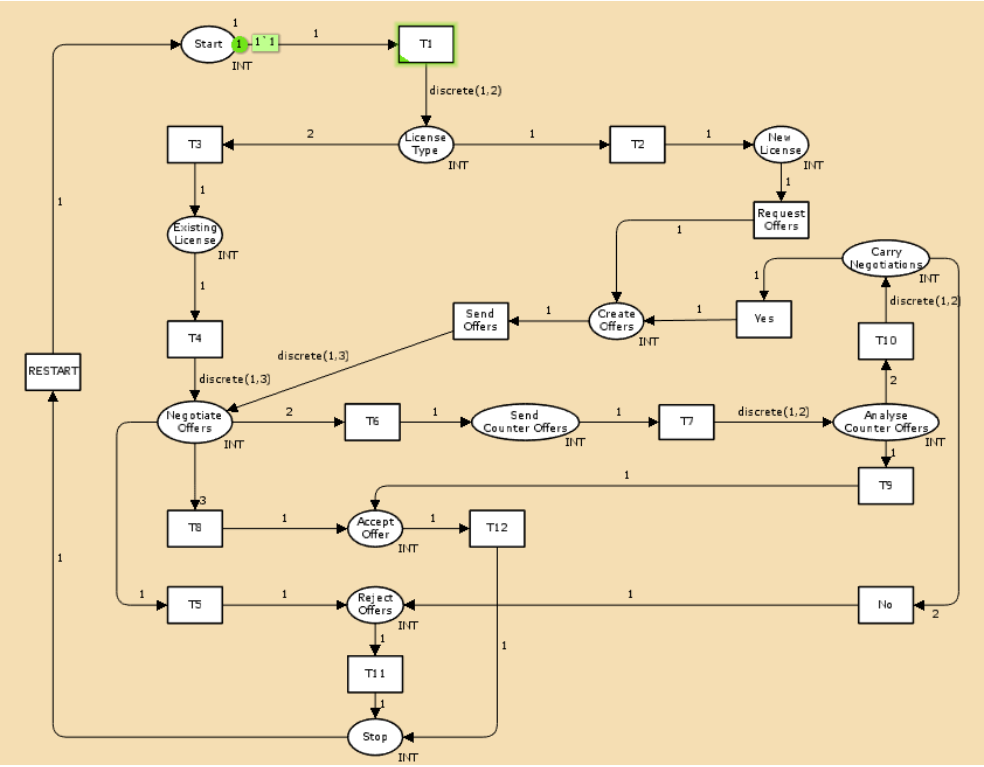


Figure 5: Bargaining Petri Net

Figure 5 shows the Petri net model for our bargaining protocol. Like our Petri net model of our bidding protocol, certain steps in the protocol can have a number of different outcomes, which we have modelled using the discrete function in CPNTools.

We have used the state space analysis component of CPNTools to analyse the net. Analysis shows

us that the Petri net is live, reachable and safe. This means that every state of the protocol is reachable with no deadlocked states. Because the net is safe, there will be no performance degradation in repetitive iterations of the protocol, and because bargaining inherently involves a number of iterations, this is an important result.

However, there is a possibility for a circular deadlock condition if the client refuses all the offers offered by the rights holder but continues to attempt to negotiate a favourable license, while the license holder continues to offer new license terms. This does create an added problem should the client be automated as it could create a potential denial of service attack. To avoid such a scenario, it could be useful to extend the protocol to keep count of the number of negotiation requests from a particular client during a particular session and stop negotiations after a predetermined number of negotiation cycles, but such decisions would depend on the business logic and the scenario of the negotiations.

7 ODRL v2 as a Language for Negotiation

In [26], the authors discussed a logic based language for negotiation between two parties, which is described in table 7.

Illocution	Meaning
request(i,j,φ)	a request from *i* to *j* for a proposal based on φ
offer(i,j,φ)	a proposal of φ from *i* to *j*
accept(i,j,φ)	*i* accepts a proposal φ made by ajent *j*
reject(i,j,φ)	*i* rejects a proposal φ made by ajent *j*
withdraw(i,j)	i withdraws from negotiation with j

Table 1: Illocutions for a logic based negotiation language [26]

In ODRL v2 [15], φ is the main body of the language, consisting of *permissions* and the affected *assets*. Parties *i* and *j* are handled by the *party* element. ODRL v2 has also extended the functionality presented above with an *agreement* illocution, which is an agreement for a use license φ between *i* and *j*:

$$agreement(i,j,\varphi)$$

Another illocution, *tender* has been proposed, but yet to be accepted and integrated into the language. This will allow for the expression of tenders, to cater for the bidding protocol. The current draft of the ODRL-v2 model is shown in figure 6.

The *agreement* and *offer* elements were the only possible state in ODRL 1 [14], while XrML (and subsequently MPEG-REL) only offers agreement in the rights expression language [2, 18]. It is sig-

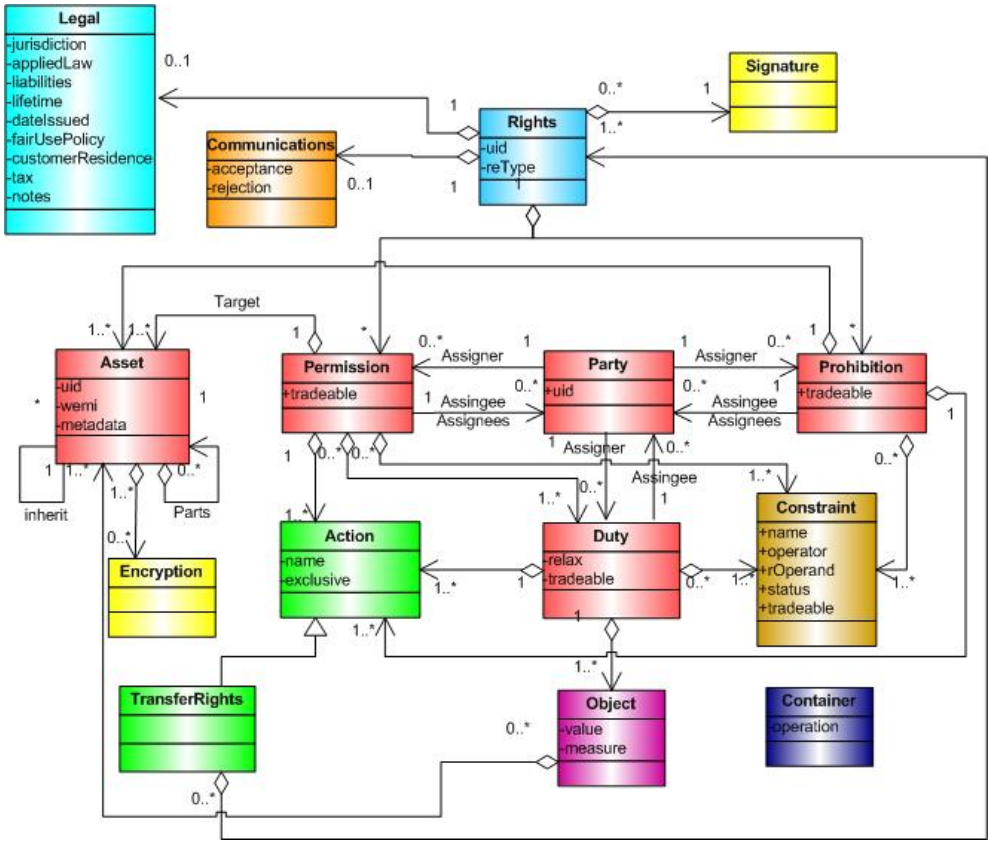


Figure 6: Latest draft of the ODRL v2 Model [15]

nificant to note that expressing the full range of negotiation positions has not affected the complexity of "normal" usage, and has not affected the overall complexity of the language. Thus, ODRL v2 is suitable as a language to express negotiations in DRM systems.

7.1 Brief Examples

In this section, we give a brief, simple example of a bargaining encounter. Due to space constraints, we have stripped out the namespace details after the first example. The schema for this example has been posted on ODRL v2 mailing list by the authors.

7.1.1 The Request

John would like to acquire a license for an ebook with the id data://123.456/2/23/23. John makes use of his Jabber enabled Google id[3] jabber://john@gmail.com.

```
<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license xmlns:o-v2="http://odrl.net/2.0/ODRL-V2"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:enc="http://www.w3.org/2001/04/xmlenc#"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://odrl.net/2.0/ODRL-V2
     odrl-v2-jan-2007.xsd"
   o-v2:reType="Request">

   <o-v2:uid>request://123.123/1.1</o-v2:uid>
   <o-v2:party o-v2:role="requestor">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:lifetime>PT30M</o-v2:lifetime>
      <o-v2:dateIssued>
         2007-08-23T09:00:00.000+02:00
      </o-v2:dateIssued>
   </o-v2:legal>
   <o-v2:communication o-v2:state="initial">
      <o-v2:targetAsset>
         <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
      </o-v2:targetAsset>
   </o-v2:communication>
</o-v2:license>
```

7.1.2 The Offer

The license server responds back to John with a license offer valid for 6 months costing 5 Euros, payable upfront. The license server id is ls://111.222.

```
<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license o-v2:reType="Offer">
```

[3] As far as we know, this is not a valid Google id

```
   <o-v2:uid>offer://111.222/123/456/23/23/1111</o-v2:uid>
   <o-v2:party o-v2:role="assigner">
      <o-v2:uid>ls://111.222</o-v2:uid>
   </o-v2:party>
   <o-v2:party o-v2:role="assignee">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:jurisdiction>Brussels, Belgium</o-v2:jurisdiction>
      <o-v2:appliedLaw>Belgium</o-v2:appliedLaw>
      <o-v2:lifetime>PT6M</o-v2:lifetime>
      <o-v2:dateIssued>
         2007-08-23T09:02:00.000+02:00
      </o-v2:dateIssued>
   </o-v2:legal>
   <o-v2:permission>
  <o-v2:targetAsset>
         <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
      </o-v2:targetAsset>
      <o-v2:action>
         <o-v2:name>read</o-v2:name>
      </o-v2:action>
      <o-v2:duty o-v2:relax="true">
         <o-v2:action>
            <o-v2:name>Pre-Pay</o-v2:name>
         </o-v2:action>
         <o-v2:object>
            <o-v2:measure>EUR</o-v2:measure>
            <o-v2:value>5.00</o-v2:value>
         </o-v2:object>
      </o-v2:duty>
   </o-v2:permission>
</o-v2:license>
```

7.1.3 The Counter Offer

John would also like to be able to annotate the work (and thus requires write permissions). he therefore rejects the offer, and creates a counter offer.

```
<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license o-v2:reType="Communication">
```

```
   <o-v2:uid>response://123.123/1.2</o-v2:uid>
   <o-v2:party o-v2:role="assigner">
      <o-v2:uid>ls://111.222</o-v2:uid>
   </o-v2:party>
   <o-v2:party o-v2:role="assignee">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:jurisdiction>Brussels, Belgium</o-v2:jurisdiction>
      <o-v2:appliedLaw>Belgium</o-v2:appliedLaw>
      <o-v2:lifetime>PT30M</o-v2:lifetime>
      <o-v2:dateIssued>
         2007-08-23T09:05:00.000+02:00
      </o-v2:dateIssued>
   </o-v2:legal>
   <o-v2:communication o-v2:state="reject">
      <o-v2:referenceCommunication>
         offer://111.222/123/456/23/23/1111
      </o-v2:referenceCommunication>
   </o-v2:communication>
</o-v2:license>


<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license o-v2:reType="Request">
   <o-v2:uid>request://123.123/1.3</o-v2:uid>
   <o-v2:party o-v2:role="assigner">
      <o-v2:uid>ls://111.222</o-v2:uid>
   </o-v2:party>
   <o-v2:party o-v2:role="assignee">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:jurisdiction>Brussels, Belgium</o-v2:jurisdiction>
      <o-v2:appliedLaw>Belgium</o-v2:appliedLaw>
      <o-v2:lifetime>PT30M</o-v2:lifetime>
      <o-v2:dateIssued>
         2007-08-23T09:05:00.000+02:00
      </o-v2:dateIssued>
   </o-v2:legal>
   <o-v2:communication o-v2:state="counteroffer">
      <o-v2:referenceCommunication>
         offer://111.222/123/456/23/23/1111
```

```
      </o-v2:referenceCommunication>
   </o-v2:communication>
   <o-v2:permission>
      <o-v2:targetAsset>
         <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
      </o-v2:targetAsset>
      <o-v2:action>
         <o-v2:name>read</o-v2:name>
      </o-v2:action>
      <o-v2:duty o-v2:relax="true">
         <o-v2:action>
            <o-v2:name>Pre-Pay</o-v2:name>
         </o-v2:action>
         <o-v2:object>
            <o-v2:measure>EUR</o-v2:measure>
            <o-v2:value>5.00</o-v2:value>
         </o-v2:object>
      </o-v2:duty>
   </o-v2:permission>
   <o-v2:permission>
      <o-v2:targetAsset>
         <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
      </o-v2:targetAsset>
      <o-v2:action>
         <o-v2:name>write</o-v2:name>
      </o-v2:action>
   </o-v2:permission>
</o-v2:license>
```

7.1.4 The Counter Counter Offer

The license server does not wish to give the write permission for free. It rejects John's counter offer, but proposes new terms with the write permission, and the license is valid for 1 year instead of 6 months.

```
<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license o-v2:reType="Communication">
   <o-v2:uid>response://111.222/123/456/23/23/1111.2</o-v2:uid>
   <o-v2:party o-v2:role="assigner">
      <o-v2:uid>ls://111.222</o-v2:uid>
   </o-v2:party>
```

```
   <o-v2:party o-v2:role="assignee">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:jurisdiction>Brussels, Belgium</o-v2:jurisdiction>
      <o-v2:appliedLaw>Belgium</o-v2:appliedLaw>
      <o-v2:lifetime>PT30M</o-v2:lifetime>
      <o-v2:dateIssued>
         2007-08-23T09:07:00.000+02:00
      </o-v2:dateIssued>
   </o-v2:legal>
   <o-v2:communication o-v2:state="reject">
      <o-v2:referenceCommunication>
         request://123.123/1.3
      </o-v2:referenceCommunication>
   </o-v2:communication>
</o-v2:license>


<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license o-v2:reType="Offer">
   <o-v2:uid>offer://111.222/123/456/23/23/1111.3</o-v2:uid>
   <o-v2:party o-v2:role="assigner">
      <o-v2:uid>ls://111.222</o-v2:uid>
   </o-v2:party>
   <o-v2:party o-v2:role="assignee">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:jurisdiction>Brussels, Belgium</o-v2:jurisdiction>
      <o-v2:appliedLaw>Belgium</o-v2:appliedLaw>
      <o-v2:lifetime>PT1Y</o-v2:lifetime>
      <o-v2:dateIssued>
         2007-08-23T09:07:00.000+02:00
      </o-v2:dateIssued>
   </o-v2:legal>
   <o-v2:communication o-v2:state="counteroffer">
      <o-v2:referenceCommunication>
         request://123.123/1.3
      </o-v2:referenceCommunication>
   </o-v2:communication>
   <o-v2:permission>
      <o-v2:targetAsset>
```

```
         <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
      </o-v2:targetAsset>
      <o-v2:action>
         <o-v2:name>read</o-v2:name>
      </o-v2:action>
      <o-v2:duty o-v2:relax="true">
         <o-v2:action>
            <o-v2:name>Pre-Pay</o-v2:name>
         </o-v2:action>
         <o-v2:object>
            <o-v2:measure>EUR</o-v2:measure>
            <o-v2:value>5.00</o-v2:value>
         </o-v2:object>
      </o-v2:duty>
   </o-v2:permission>
   <o-v2:permission>
      <o-v2:targetAsset>
         <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
    </o-v2:targetAsset>
      <o-v2:action>
         <o-v2:name>write</o-v2:name>
      </o-v2:action>
      <o-v2:duty o-v2:relax="true">
         <o-v2:action>
            <o-v2:name>Pre-Pay</o-v2:name>
         </o-v2:action>
         <o-v2:object>
         <o-v2:measure>EUR</o-v2:measure>
            <o-v2:value>5.00</o-v2:value>
         </o-v2:object>
      </o-v2:duty>
   </o-v2:permission>
</o-v2:license>
```

7.1.5 Acceptance

John agrees with the new terms.

```
<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license o-v2:reType="Communication">
   <o-v2:uid>response://123.123/1.4</o-v2:uid>
```

```
   <o-v2:party o-v2:role="assigner">
      <o-v2:uid>ls://111.222</o-v2:uid>
   </o-v2:party>
   <o-v2:party o-v2:role="assignee">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:jurisdiction>Brussels, Belgium</o-v2:jurisdiction>
      <o-v2:appliedLaw>Belgium</o-v2:appliedLaw>
      <o-v2:lifetime>PT30M</o-v2:lifetime>
      <o-v2:dateIssued>
         2007-08-23T09:09:00.000+02:00
      </o-v2:dateIssued>
   </o-v2:legal>
   <o-v2:communication o-v2:state="accept">
      <o-v2:referenceCommunication>
         offer://111.222/123/456/23/23/1111.3
      </o-v2:referenceCommunication>
   </o-v2:communication>
</o-v2:license>
```

7.1.6 Agreement

The license server generates the final agreement for John. As can be seen in the examples, there is very little difference between expressing negotiations, and the final agreement, and thus expressing negotiations using a rights expression language does not necessarily increase the complexity of the language.

```
<?xml version="1.0" encoding="UTF-8"?>
<o-v2:license o-v2:reType="Agreement">
   <o-v2:uid>license://111.222/123/456/23/23/1111</o-v2:uid>
   <o-v2:party o-v2:role="assigner">
      <o-v2:uid>ls://111.222</o-v2:uid>
   </o-v2:party>
   <o-v2:party o-v2:role="assignee">
      <o-v2:uid>jabber://john@gmail.com</o-v2:uid>
   </o-v2:party>
   <o-v2:legal>
      <o-v2:jurisdiction>Brussels, Belgium</o-v2:jurisdiction>
      <o-v2:appliedLaw>Belgium</o-v2:appliedLaw>
      <o-v2:lifetime>PT1Y</o-v2:lifetime>
```

```
   <o-v2:dateIssued>
      2007-08-23T09:012:00.000+02:00
   </o-v2:dateIssued>
</o-v2:legal>
<o-v2:communication o-v2:state="acknowledge">
   <o-v2:referenceCommunication>
      request://123.123/1.4
   </o-v2:referenceCommunication>
</o-v2:communication>
<o-v2:permission>
   <o-v2:targetAsset>
      <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
   </o-v2:targetAsset>
   <o-v2:action>
      <o-v2:name>read</o-v2:name>
   </o-v2:action>
   <o-v2:duty o-v2:relax="true">
      <o-v2:action>
         <o-v2:name>Pre-Pay</o-v2:name>
      </o-v2:action>
      <o-v2:object>
         <o-v2:measure>EUR</o-v2:measure>
         <o-v2:value>5.00</o-v2:value>
      </o-v2:object>
   </o-v2:duty>
</o-v2:permission>
<o-v2:permission>
   <o-v2:targetAsset>
      <o-v2:uid>data://123.456/2/23/23</o-v2:uid>
 </o-v2:targetAsset>
   <o-v2:action>
      <o-v2:name>write</o-v2:name>
   </o-v2:action>
   <o-v2:duty o-v2:relax="true">
      <o-v2:action>
         <o-v2:name>Pre-Pay</o-v2:name>
      </o-v2:action>
      <o-v2:object>
      <o-v2:measure>EUR</o-v2:measure>
         <o-v2:value>5.00</o-v2:value>
      </o-v2:object>
   </o-v2:duty>
```

```
   </o-v2:permission>
</o-v2:license>
```

8 Requirement Analysis

In section 4, we discussed the various requirements for negotiations. In this section, we examine how well these requirements are satisfied by our protocols.

8.1 Raiffa's Factors Affecting Negotiation

8.1.1 The number of parties

Our protocol enables negotiations between any two parties in the DRM value chain as identified in [6] and [10].

8.1.2 Parties negotiating on behalf of a group

ODRL v2 allows for multiple parties to be represented in a use license, thus allowing for multiple rights holders and multiple end users.

8.1.3 The number of terms being negotiated

The protocol is not dependent on the number of terms involved.

8.2 Satisfying DMP Requirements

8.2.1 End-Users can express their agreement or disagreements with proposed License terms

Both end users and rights holders can express their agreement or disagreement over proposed license terms. This is handled through the "Acceptance" and "Rejection" elements in the ODRL license.

8.2.2 The Protocol shall support changes to any parameter of the License

Because the protocol makes use of the full ODRL v2.0 specifications, any aspect of the license can be negotiated.

8.2.3 The Protocol shall support automatic negotiation of license terms

The protocol does not prescribe how negotiation decisions are arrived at. Thus, it is possible for either agents or humans to make the negotiation decisions for both end users and rights holders.

8.2.4 At every step a human readable license must be provided

An ODRL license can easily be transformed into a human readable licenses, and techniques have already been developed for ODRL v1.0 for this purpose [19].

8.2.5 The protocol shall enable the setting of certain parameters as non subject of negotiation

ODRL v2.0 elements have a "tradable" attribute which can specify which parameters are non negotiable.

8.2.6 The protocol shall allow the determination of the degree of confidentiality (no eavesdrop) of the protocol

The protocol can easily be run through a secure communication tunnel (example SSL) or through encrypted SOAP messages. The level of security can be determined by individual systems concerned.

8.2.7 The protocol shall not require revealing the real identities until the protocol has been successfully concluded

The identity of the rights holder is always required, but there is no such requirement for the end user. In fact, ODRL allows for totally anonymous end user licenses. The protocol does not require end user identities, but end user identity could help in negotiation decisions (for example, frequent customers getting better deals).

9 Conclusion and Future Work

Negotiation is a crucial component of the licensing process, but no current DRM systems support licensing. In this paper we have detailed two negotiation protocols and motivated their correctness and completeness through the use of Petri net modelling. We have also motivated the use of ODRL v2 as a language to express negotiation terms, and thus reduce the need for additional translation between the language of negotiation and the use license. Thus, we have presented two of four required components for automatic, electronic negotiations as discussed in literature, and our protocol supports the inclusion of the other two components. In the future, we hope to complete the full complement of components, and we are currently investigating various strategies for agent negotiation.

10 Acknowledgements

We would like to thank the members of the ODRL v2 working group and the attendees of the 10^{th} General Assembly of the Digital Media Project for their comments and suggestions.

This work is partially supported through grants from the University of Cape Town (UCT) Council and the National Research Foundation (NRF) of South Africa. Any opinions, findings, and conclusions or recommendations expressed in this paper/report are those of the author(s) and do not necessarily reflect the views of UCT, the NRF or the trustees of the UCT Council.

References

[1] ARNAB, A., AND HUTCHISON, A. Extending ODRL and XrML to enable Bi-directional communication. Departmental Technical Report, No. CS04-28-00, University of Cape Town, 2004.

[2] ARNAB, A., AND HUTCHISON, A. Extending ODRL to Enable Bi-Directional Communication. In *Proceedings of the 2nd International ODRL Workshop* (2005).

[3] ARNAB, A., AND HUTCHISON, A. Fairer usage contracts for DRM. In *Proceedings of the fifth ACM Workshop on Digital Rights Management, Co-Located with ACM CCS 2005* (2005), R. Safavi-Naini and M. Yung, Eds., ACM, pp. 1 – 7.

[4] ARNAB, A., AND HUTCHISON, A. Piracy and content protection in the broadband age. In *Proceedings of the South African Telecomunication Networks and Applications (SATNAC) Conference 2006* (2006).

[5] BARTOLINI, C., PREIST, C., AND KUNO, H. Requirements for automated negotiation. In *The W3C Workshop on Web Services: Position Papers* (2001).
URL: http://www.w3.org/2001/03/WSWS-popa/paper19.

[6] BARTOLINI, F., CAPPELLINI, PIVA, A., FRINGUELLI, A., AND M, B. Electronic Copyright Management Systems: Requirements, Players and Technologies. In *Proceedings of the Tenth International Workshop on Database and Expert Systems Applications* (1999), IEEE, pp. 896–899.

[7] BAUSE, F., AND KRITZINGER, P. S. *Stochastic Petri Nets – An Introduction to the Theory*. Vieweg & Sohn Verlagsgesellschaft mbH, 1996.

[8] BINMORE, K., AND DASGUPTA, P. Nash bargaining theory: An introduction. *The Economics of Bargaining* (1987).

[9] CAMP, L. J. DRM: doesn't really mean digital copyright management. In *Proceedings of the 9th ACM conference on Computer and communications security* (2002), ACM.

[10] CHIARIGLIONE, L. A walkthrough in the DMP Phase II specification. *Digital Media Project* (2006).
URL: http://www.dmpf.org/documents/walkthrough_in_idp-2.htm last accessed: 2006-06-04.

[11] COYLE, K. Right Expression Languages, A report for the Library of Congress. Tech. rep., Library of Congress, USA, 2004.

[12] DIGITAL MEDIA PROJECT (DMP). Requirements for new or extended idp-3 technologies, 2006.
URL: http://www.dmpf.org/open/dmp0661.doc.

[13] ELFATATRY, A., AND LAYZELL, P. Negotiating in service-oriented environments. *Communications of the ACM 47*, 8 (2004).

[14] IANNELLA, R., Ed. *Open Digital Rights Language (ODRL) 1.1*. IPR Systems Pty Ltd., 2002.
URL: http://odrl.net/1.1/ODRL-11.pdf.

[15] IANNELLA, R., AND GUTH, S., Eds. *ODRL V2.0 - Model Semantics*. 04 May 2006.
URL: http://odrl.net/2.0/WD-ODRL-Model.html last accessed: 2006-06-05.

[16] JAMKHEDKAR, P. A., AND HEILEMAN, G. L. DRM as a Layered System. In *Proceedings of the Fourth ACM Workshop on Digital Rights Management* (2004), A. Kiayias and M. Yung, Eds., ACM, pp. 11 – 21.

[17] JENNINGS, N., FARATIN, P., LOMUSCIO, A., PARSONS, S., WOOLDRIDGE, M., AND SIERRA, C. Automated negotiation: Prospects, methods and challenges. *Journal of Group Decision and Negotiation 10*, 2 (2001), 199 – 215.

[18] MULLIGAN, D., AND BURSTEIN, A. Implementing Copyright Limitations in Rights Expression Languages. In *Proceedings of the 2002 ACM workshop on Digital Rights Management* (2002), ACM.

[19] PEIG, E., AND DELGADO, J. Embedding ODRL statements in dublin core. In *Proceedings of the 2nd International ODRL Workshop* (2005).

[20] PRUITT, D. G. *Negotiation Behaviour*. Academic Press Inc, 1981.

[21] RAIFFA, H. *The Art and Science of Negotiation*. Harvard University Press, 1982.

[22] SHARROCK, R. *Business Transactions Law*, sixth ed. Juta & Co, LTD, 2002.

[23] STALLINGS, W. *Network Security Essentials – Applications and Standards*, international second ed. Prentice Hall, 2003.

[24] SU, S. Y., HUANG, C., HAMMER, J., HUANG, Y., LI, H., WANG, L., LIU, Y., PLUEMPITIWIRIYAWEJ, C., LEE, M., AND LAM, H. An internet-based negotiation server for e-commerce. *The VLDB Journal 10*, 1 (2001), 72–90.

[25] VAN VEENEN, J., AND PRAKKEN, H. A verifiable protocol for arguing about rejections in negotiation. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems* (New York, NY, USA, 2005), ACM Press, pp. 1165–1166.
http://doi.acm.org/10.1145/1082473.1082675.

[26] WOOLDRIDGE, M., AND PARSONS, S. Languages for negotiation. In *Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI-2000)* (2000), W. Horn, Ed., John Wiley & Sons.
http://citeseer.ist.psu.edu/wooldridge00languages.html.