A Contract and Rights Management Framework Design for Interacting Brokers

Susanne Guth, Bernd Simon, Uwe Zdun
New Media Lab, Department of Information Systems
Vienna University of Economics and Business Administration, Austria
{susanne.guth|bernd.simon|uwe.zdun}@wu-wien.ac.at

Abstract

Contract and rights management – and thus property rights protection – has gained increasing importance as a quality standard in brokerage and electronic commerce environments. Contract and rights management provides information on the legal relationships associated with digital assets, as well as intellectual property rights protection and the enforcement of rights. However, many brokerage and e-commerce platforms currently in operation were not originally designed to support contract and rights management. In this context, we identify open issues in digital contract and rights management and present a framework design to resolve these issues. This framework uses standardized XML-based rights expression languages, reuses an existing role-based access control component for rights enforcement, and is extensible with value-added service components for rights management. The reference project for our work is a (heterogeneous) P2P network of interacting brokerage platforms for learning resources.

1. Introduction

The Internet is increasingly being used for the distribution of digital goods, including digital versions of books, articles, music, films, pictures, tutorials, and other products. This phenomenon has given rise to a need for intellectual property rights (IPR) protection and therefore a rights management framework. Distributed brokerage platforms organized as peer-to-peer (P2P) networks, for example, accelerate the process of adopting digital goods via the Internet and thus present new challenges to rights management.

Rights management enables the protection and enforcement of rights specified for a digital product. Digital *contract management* supports the phrasing, storage, processing, management, editing, revocation and export of contracts. Digital contracts electronically provide electronic definitions of legally enforceable terms and conditions in interactions between people and organizations. Thus digital contract management is a key technical ingredient in rights management.

Rights expression languages such as the Open Digital Rights Language (ODRL) [12] or the Extensible Rights Markup Language (XrML) [23] provide language concepts and vocabulary for the composition of electronic contracts. Most current rights language standards use standard content formats such as XML and provide an extensible core set of semantics and vocabulary.

In this paper, we first identify the motivation behind our work by presenting a typical usage scenario from the educational domain: Learning resources are offered to various parties by different brokerage platforms, and these brokerage platforms can exchange and access the services in a P2P network. We proceed to describe recurring open issues that we have identified in the context of integrating a contract and rights management engine (CRME) with interacting brokers. In the ensuing sections, we will go through the proposed integration of a rights and contract management framework into the brokerage platforms. In the process, we describe three main design steps: The identification of contract components and contract structure with regard to the particular domain, contract representation in XML-based rights expression languages, and the design and functional integration of a contract and rights management engine into a brokerage platform. We will illustrate our solution with examples from the educational domain. The framework utilizes the rights expression language ODRL for contract description and a role-based access control (RBAC) mechanism [7] for rights enforcement.

2. Educational Brokerage via P2P-Based Contract and Rights Management

In this paper, we apply contract and rights management to the educational domain. Systems which support the exchange of learning resources are referred to as educational brokers [11]. Early educational brokers comprise a central catalog with hyperlinks to remotely stored educational material. This kind of educational broker does not make exchange processes transparent. Learning resources can be downloaded anonymously from a web server without a contract arising between the provider and consumer. This architecture is not suitable for establishing business relationships between learning resource providers and consumers. Moreover, the learning resource metadata which might be available at the provider's site is not reused.

The Universal Brokerage Platform [1] is an educational broker which addresses precisely these shortcomings. This environment is built around a central brokerage platform which interfaces with various kinds of learning resource delivery systems. Universal supports a contract-based exchange process in which providers specify the terms and conditions under which they wish to exchange their learning resources, and consumers are requested to comply with these rules before accessing the learning resources. The conditions under which the learning resource is made available are then described (offer placement). Two default license agreements are offered. In addition to these predefined licenses, users can specify custom licenses, and groups of institutions can introduce their own license agreements. Based on the learning resource description and offer information, consumers who wish to access a particular learning resource are asked to agree to the terms specified (contract conclusion). They thus enter into a contract, which is stored in a booking log. Based on this contract, users can access the learning resources (resource delivery). Universal provides an RDF/XML interface for the reuse of learning resource descriptions.

However, the Universal Brokerage Platform still faces two important challenges:

- *Offer and Contract Processing:* Rights enforcement [10] is carried out at the delivery servers in a proprietary manner. As a consequence, the contracts (i.e., the rights information) can not be exchanged with other systems.
- *Scalability:* The system does not scale up conveniently because each delivery system needs to be registered manually at the platform.

The Elena project [2] addresses the latter bottleneck. At the time of writing, Universal is being extended with an interface to the Edutella P2P environment [19]. Edutella aims to specify, design and implement an RDF-based P2P infrastructure using Sun's JXTA for the exchange of learning resources. The P2P framework enables an installation of an educational broker to find and connect to all systems available within the Edutella peer group (other educational brokers as well as other systems). For its delivery system, an educational broker serves as a gateway to the P2P network, but all other kinds of content repositories can connect directly to the P2P network as well. Each peer announces its learning resource descriptions in the P2P environment. Unsuccessful search requests performed on one system can be forwarded to the lookup service of other peers.

However, offers and contracts are still incompatible among the peers. Once a learning resource is found, the user cannot see the accompanying licenses and has to register himself with the offering educational broker in order to proceed with the purchase. If the resource provider wants to offer resources on multiple platforms, the processes of learning resource description and offer placement also have to be performed multiple times. Additionally, it is not possible to bundle resources that are located on different platforms and offer a new packaged resource.

The goal of our work is to draft a more flexible P2P environment than the one depicted above. In such an environment, each peer should incorporate a contract and rights management engine. This engine should provide offer placement, contract conclusion and resource delivery within the P2P network of interacting brokers, allowing users to connect to any of the brokers and then transparently access the distributed resources. Consumers should be able to purchase resources or contingents of resource types available in the P2P network of interacting brokers rather than only the specific resources of a specific broker. In this context, it is a central requirement that each educational broker is able understand the contracts issued by other brokers in the network.

One important requirement for the aforementioned cooperation of educational broker peers is a trusted environment within this cooperation. Each cooperating peer has to be a trusted partner. Offers and contracts are to be accepted only from trusted partners. Such a network of trust can be implemented with a public key infrastructure (which is not the focus of this paper). In this paper, we concentrate on rights management for consumer access to interacting brokers, but not on the secure rendering of resources on the consumer side (this could be enforced, for example, by using additional secure viewers).

3. Open Issues in Integrating a Contract and Rights Management

There is a set of recurring open issues which can be identified in the context of integrating a contract and rights management framework in interconnected environments (such as P2P). We describe these here in order to illustrate the requirements of our framework, but also as an initial delimitation of the realm targeted by our framework from related areas such as content management or constrained and/or context-dependent access control:

- *Contract management*: A contract management engine is required which supports the phrasing, storage, processing, management, editing, revocation and export of contracts. Contracts should be checked for inconsistencies and errors.
- *Expressing domain-specific context information:* The educational domain represents its data in a certain data model. This data model provides a means of describing the educational domain with all its attributes and relations. This data model of the educational domain has to be mapped to the data model of the rights expression languages. In terms of the relevant contract context information, the

languages' models allow the definition of only a limited subset of all relevant context information in the educational domain; thus metadata from the brokerage platform (such as resource IDs and user data) have to be combined with the contract information.

- *Integrating various rights expression languages (REL):* In many contract and rights management projects, standard rights expression languages, such as ODRL or XrML, are used. In order to exchange contracts between various interconnected peers, an open contract and rights management engine should be able to integrate these languages. Ideally, it should provide a generic architecture to support import and export interfaces for multiple RELs.
- *Contract enforcement and access control:* A vital part of contract enforcement is an access control mechanism. Thus the access control mechanism has to be integrated in the contract and rights management framework in such a way that it uses the access control mechanism (and the user authentication) to actually protect the resource and challenge a request or not. As there are many sophisticated implementations of access control mechanisms suited for the rights management domain, such as role-based access control (RBAC) mechanisms, we propose component-based reuse. Another important issue in contract enforcement is the enforceability of the specified rights.
- *Providing value-added services:* Contract enforcement is one of the primary services of a rights management framework. However, many additional services can be provided and it is highly domain-dependent which of these services are relevant. Typical other services include providing accounting and sales figures, providing information about legal relationships of learning resources, supporting IPR protection, IPR discovery, and automated license phrasing. The framework should be extensible with new services and these should be dynamically integrated as components on demand.

In the following sections, we will provide a framework design that addresses the open issues we have identified.

4. Identification of Contract Components

In order to design a context-specific contract and rights management platform, it is important to gain insight on the contracts used in a specific domain. In our analysis of the educational domain, we have identified four core components of contracts (see Figure 2):

- *Parties in education* (short: parties) represent all parties in the framework and are related to the learning resources; for example, they act in the role of creators, providers or consumers of learning resources.
- *Learning resources* represent the digital goods that are traded within the framework. They can be of any type and level of aggregation.
- *Rights* are related to parties and learning resources, as they express authorizations between parties and learning resources.
- *Revenue models* represent the types of "what" and "if" values flow between parties for the transfer of rights over learning resources.

The four core components are closely interrelated: parties own rights over certain leaning resources. The rights always represent rights over a certain asset (i.e., a learning resource). The exchange of a right over a learning resource is based on a certain revenue model, etc.

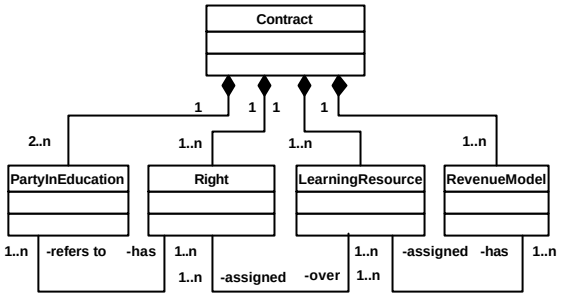

Figure 1: Basic contract model in the educational domain

On top of these basic contract components, subclasses are derived as the main contract constituents of the educational domain. Of course, this initial model can be extended and modified as new requirements arise.

As subclasses of `PartyInEducation` we identified `Learner`, `Instructor`, `Institution`, etc. and `Alliance`. `Alliance` is a Composite [8] class that may contain any number and type of parties in education.

The class `Institution` has subclasses like `Company` and `HigherEducationInstitution`.

`LearningResources` can be specialized with subclasses that each represents a different type: `EducationalActivity` or `EducationalMaterial`. Depending on the learning resource type different other attributes are supported for additional detailed description.

The required rights that may be granted to parties of learning resources are modeled as subclasses of the `Right` class, such as `PlayRight`, `PrintRight` and `ViewRight`.

Finally, various types of revenue models can be used in the educational domain, such as `Sale`, `FreeOfCharge` or `Barter`. If `Sale` is specified as the revenue model, it means that monetary transactions take place for the exchange of a learning resource. The revenue model `FreeOfCharge` stands for non-monetary transactions for the learning resource exchange. In this context, `Barter` specifies that one learning resource is exchanged in turn for another learning resource.

Our contract data model is a logical analysis view of the educational domain. Conceptually, some parts of the data model, such as parties and learning resources, are wrappers that only extract the relevant information, for example IDs, from the educational broker's data

repository . In turn, other parts of our model like rights (and possible right constraints) are providing the RBAC component with roles and rights. In order to provide the RBAC component with valuable information, the contract data model should be revised with regard to compatibility in subsequent processes. Roles in the contract data model should ideally match to roles of RBAC and granted rights should map to real access rights at the content repository. The contract and right management framework can be seen as glue between these systems that provides an implementation for our contract data model and also provides additional services associated with rights management.

In the next step we need to map the contract data model to the concepts of the rights languages to an implementation model for our contract and rights management framework. This issue is exemplified in the next section with ODRL. However, other rights languages can be used as well.

5. Contract Representation in ODRL

Rights expression languages are designed to formulate usage rights for assets as well as to express what kind of rights certain parties have over assets. Thus rights expression languages are a technical means to express digital contracts. To represent the contract model (cf. section 4) in a machine-readable and processable language, we make use of the XML-based rights expression language ODRL. Other rights expression languages like XrML are also suitable for this purpose. We chose ODRL because of the clear structure, its generality, it appropriateness for the educational domain, and its open source license.

5.1 Mapping of contract model to ORDL Language Concept

The root element in the ODRL language is the "Rights" element that represents one license. In the ODRL terminology, a contract is called an "Agreement"; the agreement element and the "Offer" element are directly connected to the rights element. An agreement comprises the elements "Party," "Asset," and "Permission". In the agreement construct, the party element represents the people that entered into a contract, e.g., the rights holder or the consumer of the asset. The asset element represents the digital product that is subject to the contract and the permission element represents the granted rights over the asset. Permission can be associated with "Requirements" and "Constraints." A requirement is a prerequisite that has to be fulfilled before a right is granted, and a constraint narrows the granted right by time, territory, user, etc.

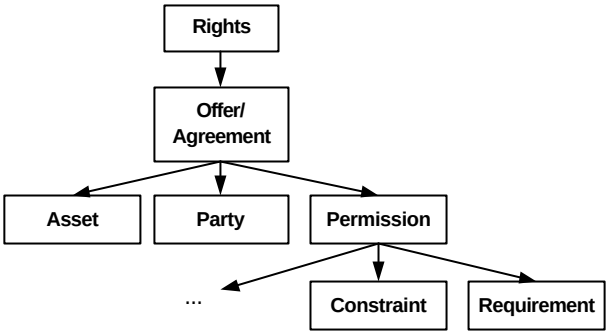

Figure 2: Subset of the ODRL language model

All named elements are illustrated in Figure 2, which shows only a subset of the ODRL language model for simplification. All elements can be further described with an ID, name, etc. in the "Context" element not shown in Figure 2.

Rights languages which meet MPEG-21 Requirements [15], such as ODRL and XrML, consist of two parts: The language concept and the vocabulary. The step just described was to map the contract components to the language concept of ODRL. When mapping very specialized subclasses to the respective rights language, the default vocabulary of the rights language might not be sufficient. The described rights languages are extensible; therefore, one is able to define a new vocabulary to reach a sufficient wealth of expression resources.

The top-level of the contract data model that we identified for the educational domain can be mapped to ODRL as follows: "Learning Resources" are described by the asset element and the "Parties of Education" can be mapped to the Party element. The contract component "Rights" is described in ODRL by the permission element that may optionally be narrowed by constraints. There is no concept in ODRL to which "Revenue Models" can be directly mapped. Revenues can be expressed in ODRL by a requirement assigned to a permission. As a requirement is a prerequisite for the specific right, this concept has the power of specifying a payment that has to be made before the desired right is granted. In Figure 3, the mapping of our contract components to the language concept of ODRL is illustrated.

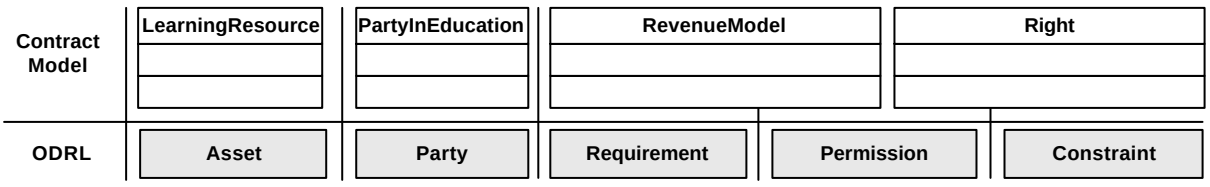

Figure 3: Mapping of contract components to ODRL language concept

5.2 Simple Example of a XML/ODRL Contract

In this section, we will give a simple example of a contract formulated in ODRL. We assume that the recording of a marketing lecture is sold to the Université Libre de Bruxelles for the price of EUR 10 with the right to play this video five times. The rights holder of the video stream is the Department of Information Systems, at the Vienna University Economics and BA. The following XML document shows the according ODRL representation of this contract. The structure of the document reflects the ODRL language concept of Figure 2. The element names can be recognized from the tag names of the language concept.

```
<rights>
 <agreement>
  <party>
   <context>
    <uid idscheme="Univ">
      urn:univ:us-wuw-deptIS
    </uid>
    <name>Department of IS, WU-Wien</name>
   </context>
   <rightsholder/>
  </party>
  <asset>
   <context>
    <uid idscheme="Univ">
      urn:univ:lr-wuw-vid-1
    </uid>
    <name>
      Marketing strategies for Universal
    </name>
   </context>
  </asset>
  <party>
   <context>
    <uid idscheme="Univ">
      urn:univ:us-wuw-uniBrux
    </uid>
    <name>
      Université Libre de Bruxelles
    </name>
   </context>
  </party>
  <permission>
   <play>
     <requirement>
       <prepay>
         <amount currency="EUR">
           10.00
         </amount>
       </prepay>
     </requirement>
     <constraint>
        <end> 5 </end>
     </constraint>
   </play>
  </permission>
 </agreement>
</rights>
```

5.3 Mapping more complex contracts to ODRL

At the brokerage platforms, we will find more complex agreements than the simple example of the previous section. We wish to take up the example of a packaged learning resource introduced in section 4. A packaged learning resource is a learning resource which comprises other learning resources. Each single learning resource may have usage rights attached to it, and additionally a set of rights can be attached to the whole packaged learning resource that applies to all components. We wish to express different permission types for different user groups (roles). For example, the packaged learning resource is offered to students for less money but in return with fewer rights. The same packaged learning resource is offered to companies for a higher price but with extended rights (see Figure 4).

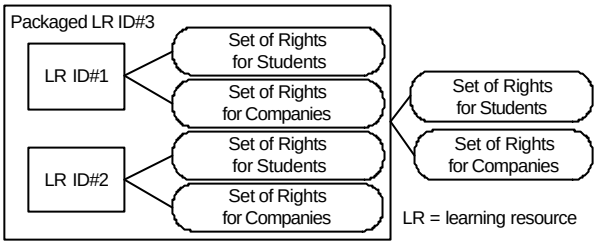

Figure 4: Illustration of a packaged learning resource

At this point, the structure of contract becomes more complex, but ODRL supports the necessary concepts to model the situation described above. ODRL provides concepts to express roles, packages, individual rights to the components in the package, as well as rights for the whole package. Our experiences show that the flexibility of ODRL might be a disadvantage in this example. There are several ways to use the ODRL concepts, which leads to an ambiguity of information. This issue has to be considered when different brokerage platform implementations exchange contracts expressed in this rights language. Thus a common interpretation concept for the ODRL constructs is required.

Now that we have mapped our contract data model to a rights expression language and formulated contracts with ODRL elements, integration with the educational broker and the underlying access control framework can be carried out.

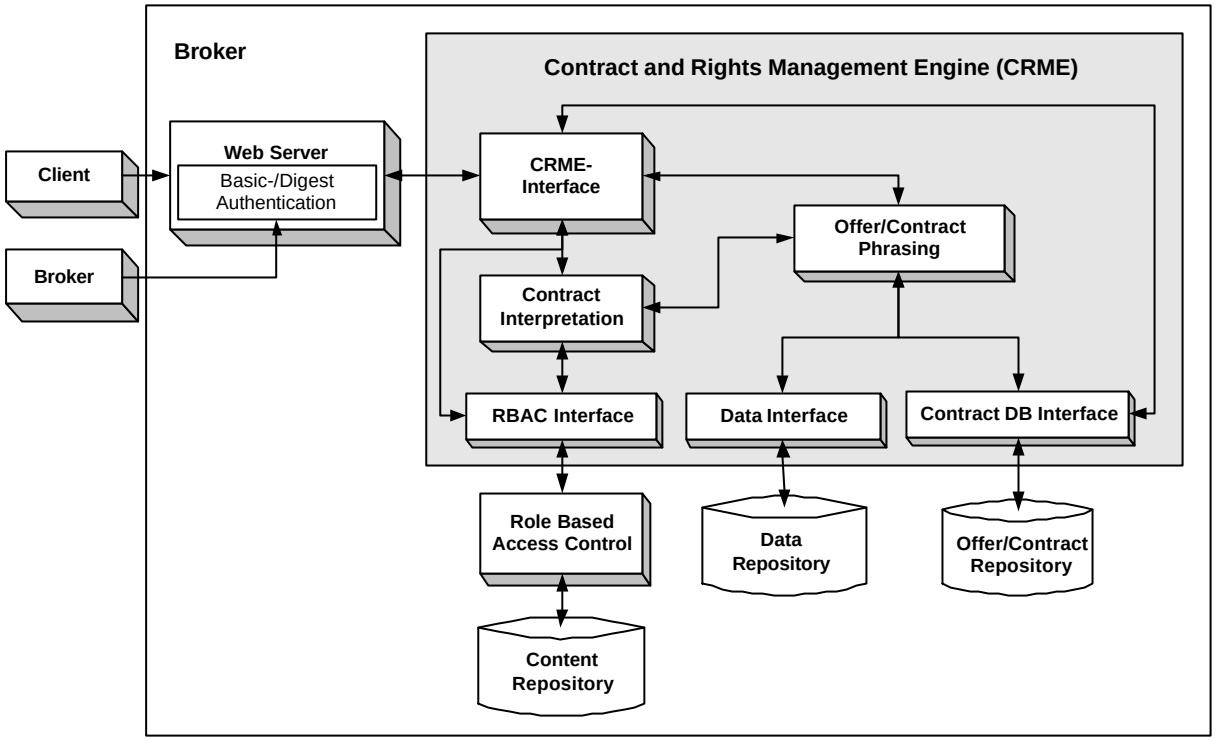

Figure 5; Components of the contract and rights management engine

6. Components of a Contract and Rights Management Framework

The contract and rights management framework consists of a contract and rights management engine (CRME) and additional components, namely the repositories and the access control mechanism. In this section, we describe all technical components and their functionalities that are essential in a contracts and rights management framework. We will describe how the contracts formulated in a rights expression language can be processed through this engine and also discuss its limitations.

The contract and rights management engine is a set of components that offer the functionality necessary for contract and rights management. It will typically operate as an additional module in a web server or application server. In general, we propose a dynamically extensible component framework, as discussed in [9]. The building blocks of the proposed engine are: the contract and rights management engine (CRME) interface, the offer and contract phrasing component, the contract interpretation component. Moreover, there are interfaces to the metadata repository, to the access control mechanism and to the repositories (see Figure 5 for an overview).

- *Contract and Rights Management Engine (CRME) Interface:* The CRME Interface handles communication with the web server. All requests from the web server to the contract and rights management engine are received by the CRME interface and then forwarded to the responsible component(s). The CRME interface also provides the export and import interface for the respective rights expression languages, such as ODRL and XrML. Export and import interfaces for rights languages can be used to exchange contracts with other brokerage platforms.
- *Offer and Contract Phrasing:* This component supports both steps of the contract creation, namely offer formulation by the content provider and conclusion of the contract by the consumer. For offer phrasing, the component needs the information of all authors and rights holders as well as the resources they provide. This information (the IDs of contract parties and resources as well as their description) is made available by the Data Repository. On the basis of this data, the rights holder can create offers for resources or for bundles of resources. Creating an offer basically means specifying usage rights and permissions for a (set of) resource(s). Permissions can be specified for an individual as well as groups or roles of individuals, e.g. “instructors.” The resulting offer is formulated in the XML-based rights expression language and stored in the offer and contract database. The component also handles the conclusion of contracts. In order to present the consumer with all offers requested, the component queries the offer and contract database for offers and provides the consumer with the detailed offer information. If the consumer accepts the usage rights as well as the terms and conditions of an offer, the component will create a new contract (formulated in a rights expression language),

implement the new access rights and store it in the offer and contract database.

- *Contract Interpreter:* The contract interpreter receives contracts from either the offer and contract phrasing module, the CRME interface or the offer and contract repository for interpretation. Interpretation means that the contract's usage rights, terms and conditions, formulated in the rights expression language, have to be brought into a format for further processing. For example a contract contains the usage right "print" for a resource in the content repository for a certain user. The access rights necessary to "print" the resource should be automatically implemented on the basis of the contract. The access control mechanism does not understand rights expression languages by nature, so a component is needed to translate these usage rights to methods offered by the access control mechanism's API.
- *Access Control Interface, Offer/Contract Interface, and Data Interface:* The engine interacts with various interfaces to the environment. It coordinates interaction with the data repository, the access control component and the offer and contract database. Each of the named interfaces is a separate component of the engine and provides a connection to external reused components.

The following external components are required by the rights and contract management framework. We reuse already existing and publicly available components such as databases or access control mechanisms.

- *Offer/Contract Repository:* The contracts are stored in a contract database used by the CRME via the contract database interface. In general, all external components are accessed by wrapper objects that provide the respective functionality. All offers and completed contracts are stored in the contract database with a new offer contract ID; thus, the responsibility for creating contract IDs stays within the contract and rights management engine. The engine uses the unique IDs of the Data Repository (i.e. unique, string-based identifiers) to reference parties and resources; thus, the responsibility for creating these IDs stays with the brokerage platform that maintains the Data Repository.
- *Data Repository:* The Data Repository of the brokerage platform contains data relevant for contract and rights management, such as information on parties, users and learning resources. This information is necessary for offer and contract phrasing, for example. Of course, the data repository can also be split into more than one physical database. In the case of Universal, there are two different data repositories; the user data repository and the metadata repository. The metadata repository stores the metadata of learning resources using the Resource Description Framework (RDF) [17]. In Universal, the metadata themselves are also described with standard formats; for instance, parties such as learning resource providers and learning resource consumers can be described with vCard [3]. Assets such as learning resources to which the specified rights apply can be described with the Learning Object Metadata Draft standard (LOM) [14] or with Dublin Core metadata [4]. The metadata on digital content that already exists in the brokerage platform is referenced by IDs. In this way, the relevant metadata is included in digital contracts.
- *(Role-Based) Access Control Component:* The contract and rights management engine also provides Component Wrappers for access control mechanisms. Currently, xoRBAC [20] is used for handling the details of rights enforcement on the system layer. The access control component supports the framework with processing its access control tasks. This way the framework works as glue between the P2P environments and the access control system. The access control component provides abstractions for rights and constraints; therefore, the contract interpreter extracts this information from the contracts and supplies it to the access control system. The access control system actually handles the details of rights enforcement and of applying the constraints provided.
- *Content Repository:* The Content Repository stores the resources traded on the brokerage platform on the basis of contracts. Access to the Content Repository is controlled by the access control component.

The introduced design of a contract and rights management framework currently does not address rights enforcement for resources once a resource has been delivered. However, the framework is open for extension to include this functionality. In addition, a public key infrastructure is necessary to provide the trusted environment in the P2P network.

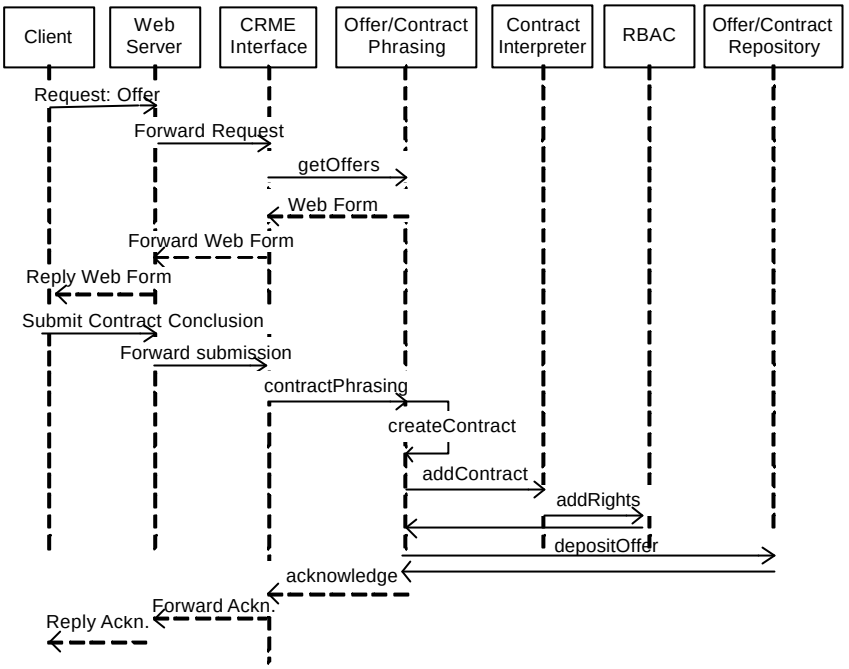

Figure 6: Sequence Diagram: Contract Conclusion, Interpretation and Storage

7. Typical CRME Scenarios

In this section, we describe concrete application scenarios to illustrate the data flow through the system. First of all, the content providers of the brokerage platform have to place offers for their content (see Figure 6). The central component that supports the offer placement is the offer and contract phrasing component. It provides the functionality for the content provider to specify usage rights for the resources he owns. Once the provider has placed an offer for one of his resources, the offer is stored in the offer and contract repository.

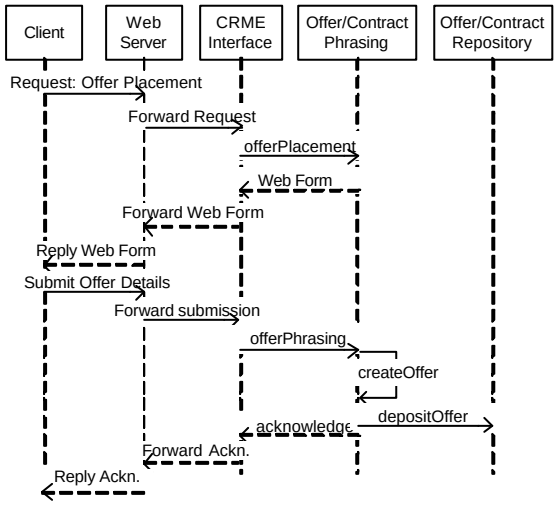

Figure 7: Sequence Diagram: Offer Placement

The next scenario is the conclusion of contract by a consumer (see Figure 7). The consumer requests to "buy" a resource offered by one of the providers. The engine provides him the functionality to choose from the offered resources. The consumer agrees to one of the offers and enters into a contract with a content provider. After the conclusion of the contract, the contract phrasing module creates a contract that is then forwarded to the contract interpretation module and to the offer and contract repository for deposit. The interpretation module interprets the specified usage rights and initiates the access control mechanism to implement these usage rights for the consumer, so that the consumer may technically have access to the resource he "bought".

The next scenario describes a registered consumer who requests access to a resource he "bought" earlier (see Figure 8). The consumer has to be authenticated of the platform by HTTP authentication methods, such as basic or digests access control. However, these implementations trigger the contract and rights management engine to challenge the request. Thus the brokerage platform sends a query to the CRME interface to determine whether access to the requested learning resource can be granted. The CRME interface initiates the role based access control (RBAC) component to evaluate the access request. If a contract exists with the particular user that comprises the permissions to access this resource, the RBAC component grants the access to the resource and allows its delivery.

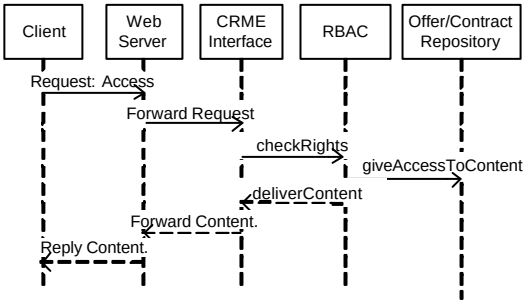

Figure 8: Sequence Diagram: Access Request

The contract and rights management engine can also handle other usage scenarios, for instance a process where the contracts are given to the consumer after conclusion. The contract can then be considered a ticket that grants certain access rights to the user, and can be executed with any arbitrarily chosen broker on the network.

8. Related Work

In our system, we use XML-based contracts to exchange processable information between different contract and rights management systems. The contracts represent an interface between the different systems; each system may interpret and process the contract content independently. Executable Trading-Partner Agreements (TPA) [22] are also a kind of contract that trading partners in electronic commerce have agreed on. These are also formulated in an XML-based language. The TPAs additionally contain policy information for different layers in the protocol stack, whereas our contracts only contain information for the application layer. The TPAs contain an agreement on functionality and services that the trading partners offer to each other. The partners agree on predefined and implement procedures that may be called by the remote trading partner rather than agreeing on usage rights over digital goods.

WebGuard [18] is also a content protection system for rights enforcement of web documents. The tool allows content owners to enforce control over distributed content to a certain extent, such as print and play control. WebGuard addresses the field of guarding content rendering after delivery to the consumer. It enforces the specified rights on the client side using common technology, namely web browser plug-ins. In contrast, our framework focuses on rights management in the context of interacting brokers. Once a client has access to a resource, no further rights enforcement is provided. Still, our framework can be extended with secure viewer technologies. However, using our framework does not require clients to download special tools (or plug-ins) to render the content.

Several commercial projects offer solutions for contracts and rights management, such as the Electronic Media Management System (EMMS) of IBM [13] or the RIGHTS|SYSTEM of InterTrust [6],[5] . Again, in contrast to our framework, both frameworks are designed on the basis of secure containers, meaning that the content is distributed in a security wrapper that can only be opened after a content viewer, a software program on the client system, has been consulted. Therefore, these solutions include additional modules for packaging, clearing, promotion, and distribution services in their basic framework. Our system primarily works with, but is not limited to, trades, where content and rights are kept separately from each other. The access rights of the content are enforced by the interacting brokers directly and not by client software on the consumer's machine. Choosing the best architecture of these two different approaches depends on the nature of the distributed content. Our system is sufficient in the educational domain and especially for regulating access to frequently-changing content, like newspapers. For content in larger, mostly anonymous communities, such as music or video exchange, additional secure viewers may be required.

The two systems aim at providing a platform for a broad range of different, widely-used content formats; however, the number of supported content formats is limited by the availability of respective viewers. As our framework does not require guarded content rendering, it is independent of the exchanged content formats. Some commercial software systems provide right management only for special content types, for instance: Microsoft's Media Player supports all content in the "Windows Media Format." Adobe's Acrobat supports rights management for PDF documents (including e-books), and Real Networks' RealOne software supports secure streamed audio and video content. IBM and Nokia a cooperating currently to develop a solution for mobile content [21]. All four systems enforce usage rights by means of software on the client machines.

Kwok and Lui [16] describe a license management model for P2P music sharing. It introduces a framework for rights management for interconnected peers. Licenses are also described with a rights expression language (XrML) to process the specified usage rights. However, in this case the exchanging peers are consumers of music (consumer-to-consumer relationship) rather than music brokers (business-to-business relationship). As licenses and viewer software are deployed on the client machine, the model faces potential problems of fraud.

9. Conclusion and Future Work

We have proposed a component-based framework design to address open issues in the design of contract and rights management. It was especially designed for interacting brokers, for example those organized in P2P networks. Thus we have provided a rights management concept and framework design suitable for the transparent exchange of resources between the brokers, but not focusing on the secure rendering of content on the consumer side.

Contracts can be exchanged using standard rights expression languages. The internal design is based on a contract data model for the specific domain (here: educational domain). Contract data models can be designed for other domains in a similar way. Subsequently the contract model of the specific domain is mapped to the concepts of a rights expression language, exemplified with ODRL. Other rights expression languages can be used as well.

Contract enforcement and access control are addressed by a contract and rights management engine that reuses an RBAC component. The engine can also be extended with additional value-added services.

Our current (and future) work focuses on implementing a more sophisticated rights expression language interpreter as well as offer and contract phrasing components with the introduced functionalities. Moreover, we plan to provide web service interfaces for the service components of the contract and rights management engine.

10. Acknowledgements

This work was supported by the Universal project and was partly sponsored by the European Commission (IST-1999-11747). The authors would like to express their special gratitude to the following persons who actively participated in the discussion of the contract and rights management framework and rights expression languages: Gustaf Neumann, Mark Strembeck, and Renato Ianella.

11. Bibliography

[1] Brantner, S., Enzi, T., Guth, S., Neumann, G. Simon, B., "UNIVERSAL - Design and Implementation of a Highly Flexible E-Market Place of Learning Resources". *In Proceedings of the IEEE International Conference on Advanced Learning Technologies*, Madison, IEEE Computer Society, 2001.

[2] Brantner, S., Henze, N., Gunnarsdottir, S., Kieslinger, B., Klobucar, T., Kuechler, T., Nejdl, W., Neumann, G., Quemada, J., Siberski, W., Simon, B. Vrabic, G., "ELENA - Creating a Smart Space for Learning". *In Proceedings of the 1st International Semantic Web Conference*, Sardinia, Italy, 2002.

[3] Dawson, F., Howes, T., *"vCard MIME Directory Profile"*. The Internet Society, http://andrew2.andrew.cmu.edu/rfc/rfc2426.html, 1998.

[4] Dublin Core Metadata Initiative, *"Dublin Core Metadata Element Set, Version 1.1"*. http://dublincore.org/docuements/dces/, 2001.

[5] Duhl, J., Kevorkian, S., *"Understanding DRM Systems"*. InterTrust Inc., White Paper, http://www.interstrust.com, 2001.

[6] Feigenbaum, J., Freedman, M. J., Sander, T. Shostack, A., "Privacy Engineering for Digital Rights Management Systems". *In Proceedings of the ACM Workshop on Security and Privacy in Digital Rights Management*, 2001.

[7] Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn, R. Chandramouli, R., "Proposed NIST Standard for Role-Based Access Control". *ACM Transactions on Information and Systems Security*, 4 (3), 2001.

[8] Gamma, E., Helm R., Johnson, R. Vlissides, J.-M., "*Design Patterns: Elements of Reusable Object-Oriented Software"*. Addison Wesley Professional, 1995.

[9] Goedicke, M., Neumann, G., Zdun, U., "Design and Implementation Constructs for the Development of Flexible, Component-Oriented Software Architectures". *In Proceeding of the Second International Symposium on Generative and Component-Based Software Engineering*, Erfurt, Germany, 2000.

[10] Guth, S., Koeppen, E., "Rights Enforcement for Learning Media". *In Proceedings of the IEEE International Conference on Advanced Learning Technologies*, Kazan, Tartastan (Russia), 2002.

[11] Hämäläinen, M., Whinston, A. B. Vishik, S., "Electronic Markets for Learning: Education Brokerage on the Internet". *Communications of the ACM*, 39 (6), 51-58, 1996.

[12] Iannella, R., *"Open Digital Rights Language (ODRL) Specification 1.1"*. http://odrl.net/1.1/ODRL-11.pdf, 2002.

[13] IBM, *"The Electronic media Management System (EMMS)"*. White Paper, 2002.

[14] IEEE Learning Technology Standards Committee (LTSC), *"Draft 6.4 of the Learning Object Metadata (LOM)"*. http://ltsc.ieee.org/wg12/index.html, 2002.

[15] ISO, *"MPEG-21 Requirements for a Rights Data Dictionary and a Rights Expression Language"*. http://mpeg.telecomitalialab.org, 2001.

[16] Kwok, S. H. S.M., L., "A License Management Model for Peer-to-Peer Music Sharing". *International Journal of Information Technology & Decision Making*, 1 (3), 541-558, 2002.

[17] Lassila, O., Swick, R., *"Resource Description Framework (RDF) Model and Syntax Specification"*. W3C Recommendation: REC-rdf-syntax-19990222, http://www.w3.org, 1999.

[18] Mourad, M., Munson, J., Nadeem, T., Pacifici, G., Pistoia, M. Youssef, A., *"WebGuard: A System for Web Content Protection"*. IBM, White Paper, 2001.

[19] Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Ambjörn, N., Nilsson, M., Palmer, M. Risch, T., "EDUTELLA: A P2P Networking Infrastructure Based on RDF". *In Proceedings of the 11th International World Wide Web Conference*, Hawaii (USA), 2002.

[20] Neumann, G., Strembeck, M., "Design and Implementation of a Flexible RBAC-Service in an Object-Oriented Scripting Language". *In Proceedings of the ACM Conference on Computer and Communication Security (CCS)*, Philadelphia, USA, November 2002.

[21] Nokia Mobile Phones, *"Digital Rights Management and Superdistribution of Mobile Content"*.White Paper, http://www.forum.nokia.com, 2001.

[22] Sachs, M., Dan, A., Nguyen, T., Kearney, R., Shaikh, H. Diaz, D., *"Executable Trading-Partner Agreements"*. IBM, White Paper, 2000.

[23] Wang, X., *"Extensible rights Markup Language (XrML) Specification 2.0"*. ContentGuard Inc., White Paper, http://www.xrml.org, 2001.