Requirements for a Rights Data Dictionary and Rights Expression Language

In response to ISO/IEC JTC1/SC29/WG11 N4044: "Reissue of the Call for Requirements for a Rights Data Dictionary and a Rights Expression Language"—MPEG-21, March 2001.

David Parrott

Reuters Ltd., 85 Fleet St., London EC4P 4AJ
Email: `david.parrott@reuters.com`

1 June 2001

Version 1.0

Table of Contents

Executive Summary

The essence of Rights and Obligation Management is electronic contract management. When we share content with others in any meaningful way, we usually wish to apply a set of rules which should be set out in some form of contract. The vision is for a machine readable contract covering all aspects of content sharing, including contribution into Reuters systems, channel partnerships, distribution arrangements, content consumption, re-sale, and re-use, amongst others. In fact, the list is unbounded. The vision also includes all forms of data interactions, be they simple document delivery, complex transactional systems, real-time data, streaming content, or broadcast delivery. Again, the list is boundless.

Background to DRM

Digital Rights Management has for some time been closely linked with the technique of encrypting data files and managing the distribution and application of cryptographic keys in order to limit:

1. who can access the content and
2. the manner in which access can take place.

That technique is more appropriately labelled "Digital Rights Enforcement" since it is more about enforcing rights than specifying and managing them. Moreover, even when enforcement is the goal, one might consider a whole array of implementation techniques which may or may not rely on encryption technology.

In truth, the management of rights in the digital domain is far wider than the rather restrictive case outlined above. Rights (and obligations) management touches on numerous areas close to the hearts of many companies dealing in intellectual property (IP). Laying enforcement issues to one side, the value cannot be understated of simply being able to describe in a machine readable, standard format, the requirements of an IP owner on all other participants in the value chain. Those requirements can be described, broadly, as Rights and Obligations. A right is the *most* that one can do with the IP; an obligation is the *least* that one needs to do in order to be granted the right (alternatively, obligations are what one needs to impose in a contract in order to assert rights).

Assessment of Commercial DRM Technologies

There exist in the order of 30-plus vendors of DRM technology at the time of writing. Almost without exception, the commercial offerings of those vendors centre on rights enforcement via encryption technology. The various products major on content-types delivered directly to human "eyes and ears" via Web browser (and related) technology, choosing to ignore issues surrounding machine-consumption of data. The systems are proprietary and share little. Interoperability is limited to a handful of special cases where initial attempts at common rights expression languages are made and, in reality, interoperability is achieved nowhere. It is likely that commercial DRM products will differ significantly from today's offerings once standards have been agreed upon. Today's products should not be considered a benchmark for the DRM revolution that is required to make content work on open networks such as the public Internet.

Outline of Requirements for a Standardised Rights Expression Language

A basic requirement for Rights and Obligations management systems to be successful is the ability to communicate Rights and Obligations in a standard form. Machine-readability is key to the dynamic specification of electronic contracts which is, in turn, critical to the dynamic construction of value-chains. A single Rights Expression Language should be common to all aspects of commercial activity. In that way alone, straight through rules processing is made possible. Rights and obligations can be created by different participants in the value-chain and layered upon each other. Data from different sources can be mixed freely without compromising the IP Rights of any of the rights holders.

At the same time, the rights of individuals and downstream recipients of content must be protected. Rights of access and privacy are to be treated as sacrosanct. A DRM system that removes rights, previously taken for granted, due to its heavy-handed approach to the management problem should be deemed as having failed.

1 Introduction

This document describes Reuters requirements for a Rights Expression Language and Rights Data Dictionary (RDD-REL) in response to the call for requirements [1] made by the MPEG-21 Requirements Committee. The document should in no way be taken to indicate any commercial policy on Reuters behalf.

1.1 Intended Audience

This response is intended for submission to the MPEG-21 Requirements Committee in response to the committee's call for requirements (ISO/IEC JTC1/SC29/WG11 N4044). The MPEG-21 standards committee is free to use the contents of the document as a resource to guide the design of a standard Rights Expression Language.

1.2 Aims and Scope of this Response

The primary aim of this response is to provide a resource to the MPEG-21 standards committee to guide its work in building a standard Rights [and Obligations] Expression Language. In order to ensure the resource is valuable, best efforts will be made to ensure that Reuters requirements are represented as fully and as accurately as possible. This response also serves as a reference source, defining terms where appropriate.

1.2.1 Explicitly Out of Scope

The following are explicitly out of the scope of the response:

- The response specifically avoids systems modelling and specification. Requirements should in no way be considered equivalent to system or functional specifications.

1.3 Structure of the Response

The document is divided into five major sections:

1. Introduction.
2. Reuters Perspective on Digital Rights Management.
3. Requirements for a Rights Data Dictionary and a Rights Expression Language.
4. Other areas for Consideration in Building the Standard.
5. Summary and Conclusions.

Additionally, appendices contain supporting material. Acronyms and glossary terms are defined in Appendix A. A bibliography is included in Appendix B and references to items in the bibliography are made in the traditional "square bracket" form (e.g., [1] refers to the first entry in the bibliography).

1.3.1 Hints on how to Read the Document

Sections 1, 2, 4, and 5 of the response give background information and set the context for the requirements listed in Section 3. Readers are encouraged not to skip the background sections.

Section 3 is divided into a number of subsections, in descending order of abstraction. At the most abstract, Section 3.1 deals with the structure of the standard itself. Section 3.2 deals with requirements about rights, such as the way that they are managed and structured; Section 3.3 lists requirements for specific Rights and Obligations to be addressed by the Language; Section 3.4 digs into the components used to form rights expressions; and the remaining sections expand on specific topics.

The requirements are a mixture of concrete versus abstract and specific versus generalised. Where possible, each numbered requirement describes a single criterion to be met by the standard. Some cross-linking between requirements is made under the "note" headings but, in the main, requirements should be considered individually. The breadth of subject matter covered inevitably makes navigation between requirements difficult. Each of the sub-sections within Section 3 is divided into topics. The topics provide some navigational aids through logical groupings. However, there is no implication in this grouping that requirements of different groups will be wholly unrelated.

1.4 Background

1.4.1 How the Response was Compiled

This response document contains a summary of many sources of input from Reuters. In the main, it is a collection of raw requirements organised within a logically structure. The requirements reflect the needs of various businesses and systems and are mostly grounded in the reality of how things are done today coupled with an evolutionary vision of the future. Quite deliberately, there was no attempt to rationalise or interpret requirements beyond that necessary to formulate coherent descriptions.

1.4.2 Document Status

Issued.

1.4.3 Document History

Version	Status	Circulation	Date of Issue
1.0	Issued.	MPEG-21 Requirements Committee via Rob Koenen. Reuters Internal distribution.	20010601

1.4.4 Acknowledgements

Thanks are due to the many colleagues within Reuters who contributed to this response.

1.5 Desirable Goals for the Standard

The following comments describe desirable elements of a standard such as MPEG-21. They are not hard requirements, but are included as suggested goals that might be set for the standard. It is recognised that some of these goals may already be stated elsewhere, but they are included here for completeness. The reader is also directed to Section 4 of the document, which lists considerations that might influence the standard.

1.5.1 Use of Existing Standards where Appropriate

There is a desire within the general community to "not reinvent the wheel". There are many standards in existence that, to a lesser or greater extent, touch upon the problem domain being addressed by MPEG-21 RDD-REL. One class of standards, including pre-existing rights expression languages, provides candidates for direct inclusion into MPEG-21 with appropriate modification and extension. Other existing standards are already complete in their own right and are best left unmodified and employed *by reference* within MPEG-21. Such standards will satisfy requirements of MPEG-21 (RDD-REL) without becoming part of the standard itself. Candidate standards for consideration for use by MPEG-21 in this manner include:

- P3P—the Platform for Privacy Preferences. A W3C initiative [3].
- XKMS—the XML Key Management Specification (submitted to W3C [8]).
- WfMC—the standards of the Workflow Management Coalition (see section 4.3).
- XrML—eXtensible rights Markup Language, from ContentGuard (spun-out of Xerox PARC) [9].
- ODRL—Open Digital Rights Language (IPRSystems) [4].
- ebXML—business message exchange framework [11].

Note that the above list is neither exhaustive nor is it intended to imply mandatory usage of any standards mentioned.

1.5.2 Conformance with Legislation

The intention is for the MPEG-21 (RDD-REL) standard to be deployed globally and ubiquitously. Therefore, there is an underlying assumption that it will take account of legislative constraints covering such areas as privacy, fair use, fair contracts, and data protection. The standard should also be sufficiently flexible (and provide appropriate management and change control procedures) to handle future changes in legislation.

1.5.3 Internationalisation

As a global standard, one would expect MPEG-21 (RDD-REL) to take account of such issues as legislative requirements, data representation requirements, and linguistic requirements in a global context.

1.6 Definitions and Assumptions

1.6.1 Definitions

The following definitions are valid within the scope of this requirements response document. They are not guaranteed to match definitions made elsewhere. The requirement statements are dependent upon the definitions given here for validity. Requirements should therefore not be quoted out of context. In order to maintain the correct context, the terms defined here may require translation when requirements are incorporated into other documents. Terms defined in this section will appear in Title Case in the body of the document.

1.6.1.1 The “Language”

References to and requirements specified in terms of the “Language” should be taken to indicate the combined Rights Expression Language and Data Dictionary that will result from the MPEG-21 activities. Moreover, for functionality requirements, the “Language” encompasses both an extensible core and any “Standard Prelude” as defined below.

1.6.1.2 “Digital Object”

The term “Digital Object” will be used throughout this document to describe any and all entities that can be the subject of Rights and Obligations. The intention is to make the definition of Digital Object as wide as possible. Within this document the term does not imply any one type of entity, such as a document or a finite or static collection of bytes. Examples of Digital Object may include (but are not restricted to) any of the following:

- Files
- Data packets within a network transmission
- An electronic interface to a service
- A database
- An unbounded stream of data packets (a pipeline)
- Logical entities such as logos (but not necessarily a specific electronic representation)

1.6.1.3 “Standard Prelude”

Borrowing from the terminology of at least one programming language, the term “Standard Prelude” is used to describe the default set of definitions that form the basis of an extensible system. The dictionary defines “prelude” as “*prel´ūd, n*. a preliminary performance or action: an event preceding and leading up to another of greater importance: …a short independent composition such as might be the introduction of another…”. In the context of the Rights Expression Language and Data Dictionary the Standard Prelude should provide sufficient definition to enable the Language and Dictionary to be used in many different and complex scenarios without extension. The nature of a Standard Prelude is that it has no special status over that which might have been defined using the primitive tools of the Language. The prelude itself is not part of that core. The requirements for a Standard Prelude are described in Section 3.1.1.

1.6.1.4 "Rights Language" / "Rights Data Dictionary"

Throughout this response, every effort has been made to refer to both Rights and Obligations. Although the text of the MPEG call for requirements [1] acknowledges the importance and relevance of specifying obligations, the title of the call refers solely to a "Rights Language" and a "Rights Data Dictionary". On occasion, consequently, the term "rights" appears without a corresponding "and obligations" within this response. In virtually all such cases, the intention of the author is to imply the inclusion of "obligations" in the argument.

1.6.1.5 "Contact" and "Interaction"

There are many ways in which a person or system can interact or make contact with a Digital Object. For example:

- An Object may be consumed in some way (printed, viewed, played, processed, etc) to the benefit of the consumer.
- An Object may be handled by a distributor, which may be a passive or active operation with respect to the Object's state.
- A value-chain participant may add value to the Object.

For the benefit of certain Rights and Obligations expressions, the generic and interchangeable terms "Contact" and "Interact[ion]" are used in this document to include all of the above and any other ways in which one may make contact (or interact) with an Object.

1.6.1.6 "Agent"

In this document the term "Agent" will be used to describe any entity which Interacts with a Digital Object, such that Contact with the Object is subject to rules expressed in the Rights Expression Language. Examples of Agents are consumers, distributors, contributors, computer systems (which process Objects), and network services (such as caches, and edge-servers).

1.6.1.7 "Context"

A "Context" is the situation in which an Agent makes Contact with a Digital Object. Rules will be defined in terms of Agents, Objects, and Contexts. It is often more useful (and necessary) to define rules in terms of Contexts rather than Agents. For example, a human consumer may be subject to different constraints depending on the role being adopted (e.g., a private individual versus a commercial role within a corporation, or even between the same individual adopting different commercial roles in the same corporate organisation). The role (and other information, such as the time of day, contracts in place, etc) helps determine the Context.

1.6.1.8 "Token"

For the purposes of this response, a Token is defined as a trusted electronic or physical construct which may be passed (issued) from one Agent to another. Possession of a Token may be a qualifying condition for certain Rights and Obligations rules. Tokens may be simple or complex in structure and are usually associated with some level of trust. In the main, Tokens should be difficult to forge. Examples of physical Tokens include theatre tickets, membership cards, cash, and driving licences. Examples of electronic Tokens include digital certificates, electronic cash substitutes (e.g., Beanz), digital tickets (e.g., Kerberos Tokens), and electronic coupons entitling the holder to discounts at e-commerce stores.

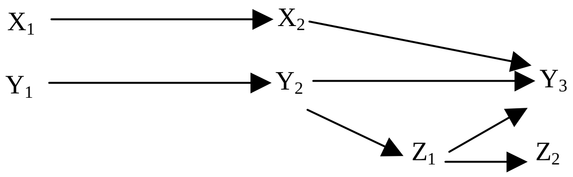
1.6.1.9 "Publisher"

The term Publisher is used throughout this document to refer to a rights owner who issues content for handling and consumption by downstream Agents. In a sense, a Publisher is an originator of content because she is responsible for unleashing it on an unsuspecting world. However, a Publisher may have taken content contributed by others (e.g., authors). Also, distributors may add value to content, thus becoming Publishers in their own right. There is therefore no special status on a Publisher. She is an Agent like all the other value-chain participants and may be subject to Rights and Obligations. If the term Publisher has any special significance, it is in the fact that a Publisher defines Rights and

Obligations over content (which may be in addition to existing rules about the same content or its subsets).

1.6.1.10 Partial and Total Orderings

A total ordering is one in which there is an absolute ranking (i.e., for a collection of items, there is a well-defined ordering between any two items within the collection). A partial ordering is one where, for a collection of items, there may items between which there is no explicit order. A total order is described diagrammatically, by a strict sequence (e.g., $x \rightarrow y \rightarrow z$). A diagram to represent a partial ordering would represent a directed acyclic graph (DAG) with potentially many starting points and many end points; e.g.:



In the example diagram, the ordering $X_1 \rightarrow X_2 \rightarrow Y_3$ is specified, as are the orderings, $Y_1 \rightarrow Y_2 \rightarrow Y_3$, $Y_1 \rightarrow Y_2 \rightarrow Z_1 \rightarrow Y_3$, and $Y_1 \rightarrow Y_2 \rightarrow Z_1 \rightarrow Z_2$. However, there is no required ordering between X_1 and any of Y_1, Y_2, Z_1, or Z_2.

1.6.2 Assumptions

1.6.2.1 Architectural Model

We assume that implementations making use of the Language will be able to support the requirements of complex value chain architectures such as the example shown below (see Figure 1). Information flows are bi-directional in the architecture, as are contractual terms and conditions. The architectural assumption is one of an object-model with all entities in the system (data, delivery channels, individuals, value-chain participants, etc) being represented in a uniform object-centric manner, thereby enabling reasoning to be expressed in terms of any and all entities within the system. For further discussion about object models, see Section 4.2. One special class of objects is the Agent class. Agents Interact with (act upon) Digital Objects (see Figure 2) and are subject to the constraints of rules expressed in the Language in terms of Agents, Objects, the Context of the Interaction, and other factors such as Tokens possessed by the Agent.

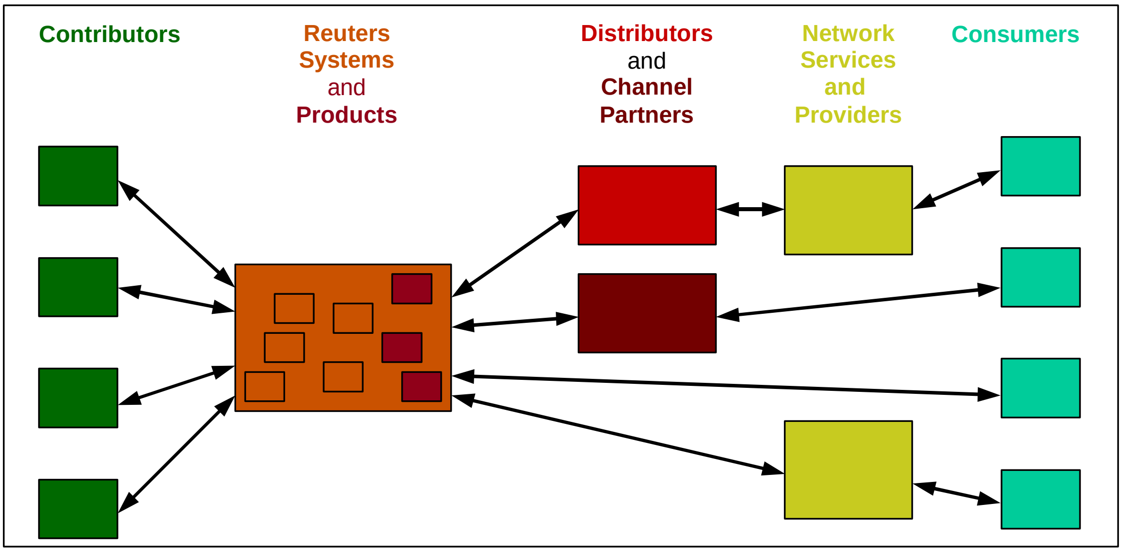

Contributors
Reuters Systems and Products
Distributors and Channel Partners
Network Services and Providers
Consumers

Figure 1 An Example of a Complex Value Chain

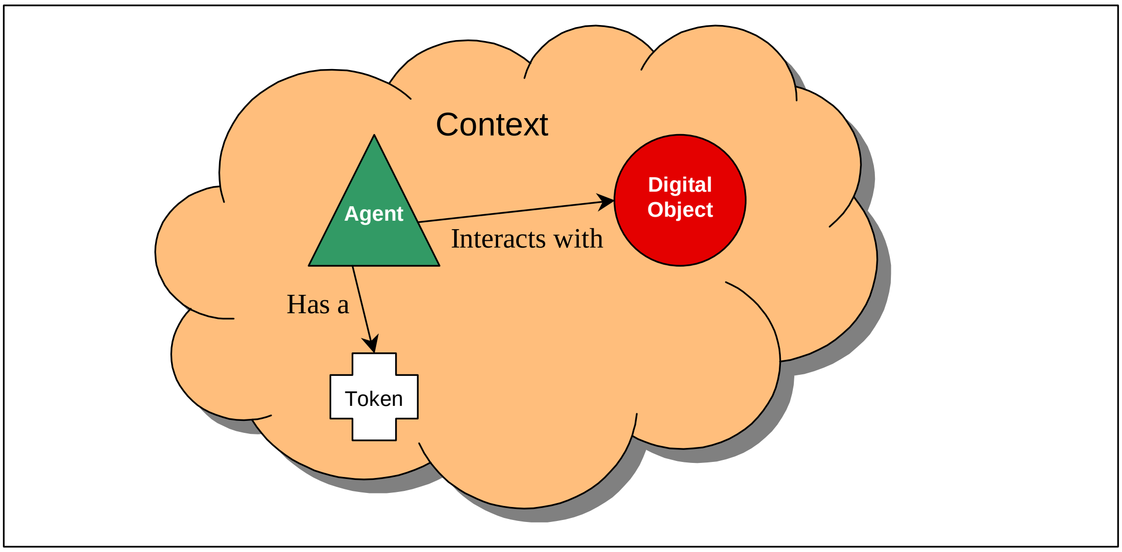


Figure 2 Architectural Elements upon which Rights and Obligations Expressions are Predicated

2 Reuters Perspective on Digital Rights Management

2.1 *Rights and Obligations*

Digital Rights Management is as much about specifying obligations in order to undertake some activity as it is about placing limitations on the scope of an activity. Examples of obligations on end-consumers are fairly straightforward and include such obligations as payment for services and use of particular applications software. However, obligations are particularly interesting in the case of a complex value chain across multiple industry players where obligations on distributors, network service providers, channel-partners, etc, are varied and complex. They may include such issues as minimum trust levels, quality of service, audit requirements, financial reporting, and so on, ad infinitum.

2.2 *DRM is not synonymous with Rights Enforcement*

Digital Rights Management is often confused with rights enforcement. Certainly the latter can be a component part of DRM, but there is no reason why it must. Rights management covers the following areas:

- Methods for binding Rights and Obligations to Digital Objects in an unambiguous manner (NB: some scheme may require the unique identification of Digital Objects, others may not)
- Support of intellectual property rights (IPR) by consciousness-raising activities, such as:
  - Licensing
  - Copyright and Trade Mark statements
  - Digital watermarking
- Auditing and tracking for illegitimate usage:
  - Auditing via tools such as Copyscan from Idioma Solutions
  - Searching the Internet using tools from companies such as Digital Integrity, Envisional, and IBNet Plc.
  - Monitoring real-time communications using tools such as those provided by ipArchive.
  - Traitor-tracing: adding consumer fingerprints to content to trace the source of "leaks"
- Trust infrastructure:
  - Authentication of value chain participants (PKI)
  - Authenticated audit trails
  - Trusted environments and applications
  - Licensing schemes (signed applications/environments)
  - Trusted third parties (for clearing, etc)
- Enforcing Rights and Obligations:
  - Encryption schemes
  - Trusted environments and applications
    - E.g., Vyou.com software for protecting the desktop
    - InterTrust's RightsSystem trusted sandbox for embedded systems
  - Interoperability:
    - Key management (XKMS)
    - Chains of trust (PKI signatures etc)
  - ACLs

2.3 *Things to be done in order to make the most of DRM*

The first step in successful Digital Rights Management is to be able to describe Rights and Obligations in a manner that is widely understood. We need:

- A taxonomy of Rights and Obligations for products and services
- A mechanism for communicating Rights and Obligations to value chain participants
- A standard Rights Expression Language for the above
- A commercial and legal framework to support the technology
- Openness and standards acceptance across the industry.

To make DRM work requires effort both:

- Technical (e.g., implementing software solutions), and
- Non-technical (e.g., business and legal policy and frameworks).

Clearly, the enforcement of Rights and Obligations will be the end-goal for most employers of DRM. However, it is useful to observe that there are three, equally important components in the enforcement equation as depicted by the three-legged stool in Figure 3.

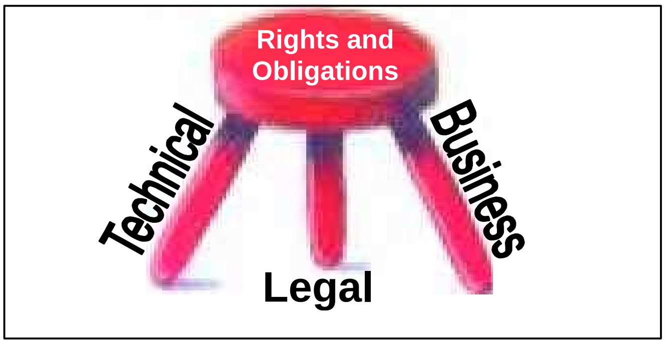


Figure 3 The Three-Legged Stool of Rights Enforcement

Additionally, the frameworks need to accommodate both extremes of:

- Lightweight digital rights models (e.g., real time, high volume electronic content) and
- In-depth rights management (e.g., deep documents with complex IP requirements).

To make the standard happen a modular structure is needed, accommodating tiered and segmented requirements. Concerns have been expressed that:

- The reach of the standard is potentially very wide.
- The needs of individual implementations are unlikely to call upon the entire scope of the standard.
- Acceptance and deployment is expected to be slow and costly. Until adoption reaches critical mass, interactions will commonly take place between three classes of Agents:
  - (i) Those who support the standard in full
  - (ii) Those who support the standard in part
  - (iii) Those who do not support the standard

All of the above require careful management for the standard to be a success.

2.4 The Essence of Digital Rights Management

The essence of Rights and Obligation Management is electronic contract management. When we share content with others in any meaningful way, we usually wish to apply a set of rules which should be set out in some form of contract. The vision is for a machine readable contract covering all aspects of content sharing, including contribution into Reuters systems, channel partnerships, distribution arrangements, content consumption, re-sale, and re-use, amongst others. In fact, the list is unbounded. The vision also includes all forms of data interactions, be they simple document delivery, complex transactional systems, real-time data, streaming content, or broadcast delivery. Again, the list is boundless.

3 Requirements for a Rights Data Dictionary and a Rights Expression Language

Requirements in this section are given in the following form according to the express request of the MPEG-21 requirements committee:

3.*x.x.x* ***Requirement title***

Requirement:

Specification of the requirement.

Note:

Additional notes.

Example:

Example in the Reuters (or other) domain.

Readers are directed to the introduction section in order to understand the scope, context, and structure of the requirements presented here. In particular, the logical progression and grouping of requirement descriptions within this section is described in section 1.3.

3.1 Requirements for the Structure of the Standard

In this section a list of requirements is given which is likely to influence structure and implementation rather than simply content within the standard.

3.1.1 Division of the Standard into an Extensible Core and Standard Prelude

Note that further requirements pertaining to the structure of data dictionaries are included in Section 3.2.7. The following requirements assume an architecture as depicted in Figure 4.

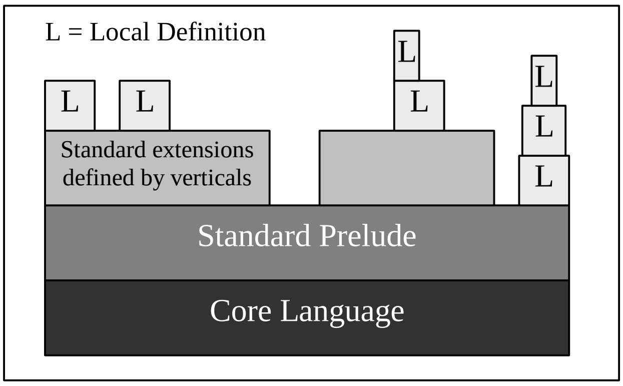


Figure 4 A Hierarchy of Definitions

The illustrated hierarchies are provided for example only and do not imply any limits on the depth or complexity of the intended hierarchy.

3.1.1.1 A Core Set of Primitives

Requirement:

A minimal core set of primitive constructs must be identified from which all other expressions can be constructed or derived.

Note:

Changes to standards are difficult and time-consuming to accomplish. The greater the level of functionality placed at the core of the standard, the higher the likelihood of the need for change.

Example:

Rather than describing specific business models in the core language, primitive expressions are provided in the core that, along with suitable hooks, would allow business model definitions to be made in the Standard Prelude (see Requirement 3.1.1.2) and elsewhere. For example, instead of defining per-per-view, rent-to-own, and other such models, the core should provide the fundamental building blocks to link obligations (e.g., payment) to actions, to allow obligations to repeat according to various criteria, provide metering syntax, and so on. The named business models have no place in the core Language. See Section 3.3.11 for an enlargement on business model requirements.

3.1.1.2 A Standard Prelude

Requirement:

The standard should be defined in terms of the extensible core and a Standard Prelude, or library, of initial extensions that define sufficient terms that the Language and Data Dictionary are useful for a large number of practical purposes.

Note:

A minimal standard is of little practical use. A Standard Prelude provides a toolset that provides practical functionality. Moreover, a Standard Prelude can be over-ridden because it has no special status. Therefore, changes can be implemented on a local basis. Global changes can be created via change-control mechanisms that preserve backwards compatibility and which may be more lightweight than the process to change the core of a standard.

Example:

Building on generalised syntax in the core, the Standard Prelude might define a basic (and probably rich) set of business models such as per-per-view, rent-to-own, price-capped-service, etc.

3.1.1.3 Enumerations

Requirement:

The core standard should not contain enumerations. It should, instead, present an open framework into which enumerations taken from the Standard Prelude (and other libraries) are plugged.

Note:

Incorporation of enumerated lists has disadvantages for extensibility. Either the list of enumerations is extended by a separate mechanism, or the standard needs to be rewritten in order to create additional enumerations. The latter option is unwieldy (and slow) while the former affords special status to those enumerated items included in the core standard. It is better to have no enumerations in the core and to extend the null standard with a set of libraries. Then, all enumerations are dealt with in a uniform manner. New enumerations can be added either by extending the standard libraries or by adding libraries.

Example:

Rendering of content by an end-consumer provides an excellent example. Assume a core standard implemented with primitive enumerations VIEW, PRINT, and SAVE. Clearly, this is display-oriented markup. The Language would needed to be extended to cope with audio data, where PLAY, CUE-BACKWARDS, CUE-FORWARDS, etc, may all be valid extensions. To handle executable code, the enumerations EXECUTE, DEBUG, REVERSE-ENGINEER, may be required. For each new data type, new rendering enumerations are required. It would make sense to partition these into separate, vertical libraries, each under individual change-control.

3.1.1.4 Extensibility

Requirement:

Extensibility must be supported throughout the Language. Hooks must be provided in the Language for the use of alternative schemas wherever sensible, with suitable attributes to allow the scheme to be identified.

Note:

Over-riding the Standard Prelude should also be seen as a natural activity. The Standard Prelude is there as a “starter-pack”; if its definitions do not suit a particular vertical segment then it is better to over-ride definitions than to use existing definitions in an unsuitable manner.

Example:

Rather than constructs of the form `<mytag value="`*foo*`"/>` it is preferable to use constructs of the form `<mytag schema="bar">`*foo*`</mytag>` where, in the latter case, *foo* can be either a simple value or a full XML construct built according to the named schema.

3.1.1.5 Promotion of Local Definitions into the Global Domain

Requirement:

Management mechanisms must be put in place to allow local definitions to be promoted into the global domain when appropriate.

Note:

This is actually a change-control requirement directed both at the core Language and the Standard Prelude. The change-control process should be made as efficient as possible to avoid divergence. Any reliance on name-spaces to disambiguate definitions should be supported by appropriate process and Language structure to allow the definitions to be merged.

Example:

The term PLAY might be defined in the Standard Prelude to refer to Interactions with video or music Objects. A games company wishes to re-use the term with special syntax to refer to controls over playing games. A local definition of PLAY is therefore introduced with attributes unique to the new domain (there are also likely to be attributes shared with the existing definitions). It eventually turns out that sufficient numbers of games services require DRM capabilities and that the local definition is being used extensively. The promotion process allows the vertical industry to vote the extended PLAY definition into the Standard Prelude, taking account of any conflicts that might arise with existing PLAY definitions.

3.1.1.6 Hierarchies of Definitions

Requirement:

A hierarchical system of arbitrarily deep and progressively localised extensions and redefinitions must be supported.

Note:

The global/local terminology used above should not be taken to indicate a two-tier hierarchy. There is a direct analogy with the addition of attributes to Digital Certificates in conformance to the X.509 Digital Certificate standard.

Example:

The Standard Prelude might define PLAY, as above. A vertical games-industry body might augment PLAY with a local definition. Sub-groups within the industry might identify particular classes of games which require unique attributes to control access. A particular games company might require additional attributes to be added to control more precisely the allowed behaviours over a particular game.

3.1.2 Inclusivity

A good standard is one that is inclusive with respect to scenarios and requirements that were not even thought about when the standard was constructed.

3.1.2.1 Reach all parts of the Value Chain

Requirement:

The standard must not make assumptions about the application of the Rights Language and Data Dictionary to specific parts of the value chain. Rather, the standard must assume a heterogeneous, multi-organisational, industry-wide value chain not limited by the commercial structure of any one class of organisation.

Note:

The standard must free itself from the mindset of publishers issuing content to end-consumers. “Customers” may be participants in a complex value chain and may never include “end-consumers”.

Example:

Examples of value-chain participants that are at least as important with respect to being subject Rights and Obligations include:

- contributors of content,
- distributors,
- channel partners,
- physical components in the communications network,
- value-added resellers.

3.1.2.2 Dataflow and commerce agnosticism

Requirement:

The standard must not preclude the specification of Rights and Obligations for any activity on the grounds that it does not fit a narrow definition of accepted dataflow.

Note:

This requirement is intended:

- Partly to highlight the needs of a cross-industry value chain;
- Partly to avoid the tendency for solutions to focus on existing concrete problems rather than the most general case; and
- Partly to reinforce the need for a single Language (or, at the very least, a coherent set of Languages) for deployment in many different, but related scenarios.

Many of the following examples are the subject of separate and more detailed requirements.

Example:

It should be possible to employ the same Rights Language for all of the following:

- To constrain how someone may contribute information into a system
- To define restrictions and obligations on distributors and channel partners
- To describe controls over any data types and any form of information flows (e.g., from straightforward data delivery to complex transactional dataflows)
- To describe how aggregators of information must behave
- To describe how value may be added to information
- To limit the mode of consumption of content by an end-user and subject them to obligations
- To limit the purpose for which content is used.

3.2 Requirements for Rights Structure and Management

This section contains a list of requirements pertaining not to definitions of particular Rights and Obligations, but at the level of managing the specifications.

3.2.1 The Relationships between Rights and Obligations

In this section requirements are expressed for the logical connectivity between rights and obligations. For maximal generality, relationships may be expressed between any combination of Rights, Obligations, and arbitrary groupings of the same.

3.2.1.1 Obligations as a Consequence of Exercising Rights

Requirement:

The Language must allow for the specification of a set of Obligations that are a consequence of exercising a Right (or set of Rights).

Note:

This is in contrast with Obligations that need to have been fulfilled before a (set of) Rights are granted (see Requirement 3.2.1.2).

Example:

The following are hypothetical examples.

- The right to print a document is granted. As a consequence of printing, the following obligations are triggered:
  - A fee of $5 is payable by the consumer to the distributor.
  - The content distributor is obligated to collect the fee, return a percentage to the publisher, to record the event within a log, and to report summary information to the publisher.
- A distributor has the right to alter news content by editing the material. As a consequence of the alteration:
  - The distributor is now obligated to add notices to the content to indicate that it is based upon an original by the publisher, but is not solely or entirely the original text
  - The distributor is further obligated to add URLs to the text to indicate where the original may be obtained for comparison.

3.2.1.2 Obligations as a Pre-Requisite for Exercising Rights

Requirement:

The Language must allow for the specification of a set of Obligations that must be fulfilled before a Right (or set of Rights) can be exercised.

Note:

This is in contrast with Obligations as a consequence of Rights being granted (see Requirement 3.2.1.1).

Example:

Acknowledgement of contractual terms via specified electronic means is a pre-requisite of being granted access to a subscription feed.

3.2.1.3 Sequencing Obligations

Requirement:

The Language must allow for the specification of a set of obligations that must be fulfilled in a total or partial temporal ordering.

Note:

See Section 1.6.1.10 for a definition of partial and total orderings.

Example:

- An example of a total ordering of obligations on a distributor is:
  1. A request for conformation must be issued back to the publisher before passing data to a particular client for the stated use
  2. On receipt of confirmation, payment must be taken from the client
  3. Data is supplied to the client across a trusted network
- An example of a partial ordering of obligations on the same distributor obtained by adding the following obligations:
  4. A minimum Quality of Service is specified for data transmissions
  5. Audit records must be returned to the publisher immediately on completion of the transaction.

  Here, steps 3 and 4 have no explicit ordering (in fact, they are coincident), so the partial ordering is 1→2→(3,4)→5.

3.2.1.4 Sequencing Rights

Requirement:

The Language must allow for the specification of a set of rights that exist according to a total or partial temporal ordering.

Note:

See Section 1.6.1.10 for a definition of partial and total orderings. Note that, a temporal ordering does not imply duration, merely a sequence. Therefore, if expiration rules are expressed according to Requirements 3.2.8.1 and 3.2.8.7 then transition conditions between the rights in the sequence can be determined independently of the ordering.

Example:

This allows for such business models as tiered access to content. This may be combined with temporal orderings for rights and obligations to achieve tiered pricing where the price off access is reduced as content gets older. For example, a real-time stock quote may cost $1 and, for the first 10 minutes of its life, be limited to people who have an exchange contract. On expiry of that rule, the quote then costs $0.5 and is available via an aggregator for the next 10 minutes. Finally access is free and publicly available. Similarly, it may be useful to determine that, in the initial 10 minutes of the lifetime a group of individual has the right to trade on the price but that this right expires beyond that time.

3.2.2 Rights and Obligations Transfer (Inheritance)

Data may be structured and linked in arbitrary ways. The requirements in this section deal with the manner in which the Rights and Obligations associated with one Digital Object can be transferred to another object by implication or otherwise. A well-known model of transfer is inheritance, typically associated with hierarchical structures. Inheritance is merely a special case of the more general concept of transfer. Requirements are specified both in the general case of inheritance and in the more general case of transfer in this section. The apparent redundancy is motivated by the needs of those who understand “inheritance” semantics versus the desire to state the most general case.

3.2.2.1 Rights Specifications for Hierarchical Containers

Requirement:

Where rights expressions can be applied at various levels within a hierarchical data container, well-defined semantics must exist to determine which rights apply to data at all points within the hierarchy.

Note:

We acknowledge that inheritance should apply to both process (e.g., business rules) and data. There is no one correct answer for the semantics for overriding rights specifications. Therefore, it is better to allow the semantics themselves to be specified as part of a rights expression, probably with default semantics where no explicit specification is given (see requirement on inheritance semantics below).

Example:

In a NewsML [2] data container, Rights and Obligations could be specified for the whole container, and once again at the level of individual news items within the container. Where conflicts arise, the NewsML case would require that the lower-level constraints apply (this is true both for weakened as well as strengthened constraints). Other container structures or delivery mechanisms may require different semantics to apply.

3.2.2.2 Inheritance Semantics

Requirement:

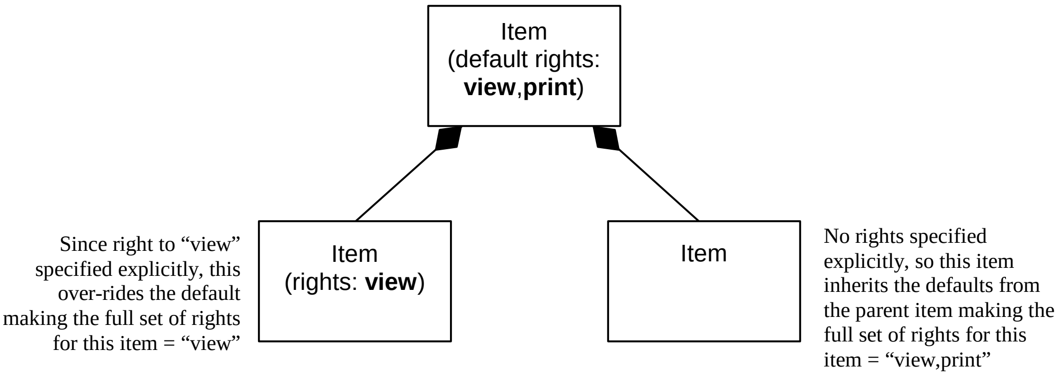
There is no one correct answer for the semantics for overriding rights specifications via an inheritance mechanism. Therefore, it is better to allow the semantics themselves to be specified as part of a rights expression, probably with default semantics where no explicit specification is given.

Note:

Default semantics might specify that lower-level rights expressions always override higher-level expressions, regardless of the strength or weakness of the constraints involved.

Example:

- In some cases it may be meaningful for the rights hierarchy to specify default rights at the higher-levels to be applied only if lower-level rights are not specified. For example, the default rights for a news item might be "view" and "print". However, if lower-level elements of the item are to disallow printing, then it should be possible to specify just "view". This would over-ride completely the higher-level rights already granted:


Item
(default rights:
view,**print**)
Since right to "view" specified explicitly, this over-rides the default making the full set of rights for this item = "view"
Item
(rights: **view**)
Item
No rights specified explicitly, so this item inherits the defaults from the parent item making the full set of rights for this item = "view,print"

- In other scenarios (for example in rights hierarchies created by successive distributions/aggregation of content) the outermost rights expressions might be allowed to strengthen, but not weaken lower-level rights specifications.

3.2.2.3 Generalised Rights and Obligations Transfer Model

Requirement:

The standard must allow for the description of arbitrary models of Rights and Obligations transfer between Digital Objects. This includes dynamic, service-based determination of transfer semantics.

Note:

For any two Digital Objects, a relationship may be expressed between them which determines how the Rights and Obligations of one of the objects affect the Rights and Obligations of the other. The Language is responsible for providing the framework to allow arbitrary expressions of this nature to be defined.

Example:

A Web page contains numerous data items. Rights and obligations are defined for the Web page. The transfer model is such that Rights and Obligations transfer from the Web page, to frames within the page, and from frames to the separate images embedded within the page (this is a typical hierarchy). Furthermore, different rules might apply to the transfer of Rights and Obligations to data items linked from this Web page using the HTTP hyper-linking mechanism, according to whether the links refer to local or remote content. Different Web pages all linking to common underlying content within a site, may transfer different Rights and Obligations to the content. This might be used, for example, as way of implementing different levels of service according to subscriptions to bronze, silver, or gold products. Each product would have its own Web page with each page conferring a different set of Rights and Obligations on content common to all products.

3.2.3 Rights and Content Independence

3.2.3.1 Rights Expressions as First-Class Entities

Requirement:

There must be provision to allow Rights and Obligations expressions to exist as first-class entities (i.e., it must be possible to make an unambiguous reference to a rights expression such that it can be referenced independently of content to which the expression applies).

Note:

This is a pre-requisite if Rights and Obligations are to be expressed over Rights and Obligations expressions (see Requirement 3.2.4.3) and for relational models where a relationship identifies both a Rights and Obligations expression and its subject (see Requirement 3.2.5.1).

Example:

A Rights and Obligations expression is to be served on demand from a dynamic Web service interface. A contract binding the dynamic expression to a content Object needs to be able to identify the expression.

3.2.4 The Types of Content over which Rights and Obligations Apply

3.2.4.1 Generalised Digital Objects

Requirement:

Examples of digital content often focus on finite data items, such as documents, images, video files, etc. Digital content is more than that. Firstly, it includes such things as infinite data streams. However, an object-oriented view of the world (rather than a data-centric view) would suggest that Digital Objects must be viewed as active, rather than passive entities (with the passive entities merely that subset of entities whose action-set is empty). For the standard to be extensible, all digital entities need to be represented. Therefore, provision must be made to encompass rights over all digital entities, not just passive data.

Note:

This is a high-level, abstract requirement. Lower-level, more concrete requirements follow.

Example:

The following are Digital Objects over which rights might be specified:

- A database accessible on a network via a query language
- A distributed object accessed via API calls (e.g., CORBA or DCOM object)
- A Web server accessed via HTTP
- A video stream from a surveillance camera.

3.2.4.2 Expressing Rights and Obligations over Derived Metadata (and Derived Data)

Requirement:

It is often possible for a downstream value-chain participant to derive metadata from source content. If the metadata is then published alongside the original content as a value-added service, this may compromise the content owner’s rights over the original data in cases where the metadata exposes too much of the underlying data or negates the need to consult the underlying data. Therefore it must be possible within the Language

1. to describe limitations on deriving metadata (or simply data) in this manner and
2. to describe a set of maximal rights and minimal obligations which can be expressed about derived data.

Note:

It is widely known that one person’s metadata is another person’s data. Where metadata is included within a content package by the originator of the underlying data, it is incumbent upon the originator to ensure that the metadata is encapsulated in such a manner that the usual mechanism for expressing Rights and Obligations can be applied. However, if metadata is derived by anyone other than the originator of the data, then a special mechanism is required to describe the rights of publication over the derived data. Note that the ability to express such rights is independent of the legal status of the rights thus expressed. For example, certain jurisdictions may grant some derived works the same status as an original work with respect to the IPR of the creator of the derived work.

This requirement is related to the addition of annotations to Digital Objects, described in Requirement 3.3.8.7.

Example:

A distributor may decide to publish synopses of news content consisting of headline, first paragraph, and highlights of the main text. The publisher subsequently discovers that access to the derived content is unrestricted and that, consequently, business is significantly affected. The publisher should be able to codify restrictions on this activity, at least to the point of requiring that the Rights and Obligations pertaining to the underlying data be placed also on the derived data.

3.2.4.3 Rights Expressions and Data Dictionaries are Digital Objects which may be subject to Rights and Obligations

Requirement:

It must be possible to specify rules about who may and may not gain access to specifications of Rights and Obligations in terms of both rights expressions and data dictionaries.

Note:

Expressions of Rights and Obligations may contain commercially, legally, or otherwise sensitive information. Therefore, it must be possible to restrict access to those expressions in the same manner as any other Digital Object. Clearly, such restrictions must be handled with care, lest all information about access be hidden. Data dictionaries may contain similarly sensitive content and should be afforded the same level of protection.

Example:

A financial markets analyst produces a report for restricted circulation. The report has associated with it a set of constraints about competitors who are excluded to a greater or lesser extent from accessing the contents of the report. The set of constraints contains sensitive information. Therefore, a top-level rights expression is created to say that, by default, nobody has access to the document unless a particular license token is held. The mechanism by which a token is issued might proceed as follows. A second rights expression (containing the sensitive competitor information) governs issuance of the token. Access to that rights expression is granted only to a specified type of licensing service. The licensing service works by reading the protected rights specification and determining whether or not the applicant has the right of access to the content. The licensing service keeps the details of the rights expression secret and issues the licence token that was the subject of the top-level rights expression.

3.2.5 Matching Rights and Obligations to Digital Objects

Rights and obligations expressions can be considered to be an example of metadata. Metadata can be associated with Digital Objects according to any one of the following three models:

1. The “self-describing” model—in which the metadata is bound tightly to the Object. An example is markup placed in-line, in and around content items.
2. The “reference” model—in which metadata and Digital Objects are separate but can be associated by reference. Associations might be achieved in any of the following ways:
   i. A reference embedded in the metadata identifies the Object to which the metadata refers
   ii. A reference embedded in the Object identifies its associated metadata
   iii. A relational model in which a relation entity contains references to both a Digital Object and metadata.
3. The “service” model—where associations between metadata and underlying Objects can be determined dynamically. Moreover, queries upon dynamic services can obviate the need to reveal the metadata. In the case of rights expressions, for example, a service may simply return a “yes” or “no” answer to an access query (or something more complex—see Requirement 3.2.9.3), rather than returning the full rights expression. Examples of the Web Services paradigm are given in [5]. (See also, Section 3.2.11, in which requirements for expression-evaluation are stated.)

3.2.5.1 Electronic Contracts

Requirement:

An electronic contract may wish to bind together numerous rights and data entities in a relational manner. Provision must exist in the standard to support this.

Note:

One can imagine that an electronic contract might be a static expression written in the Language, or might be realised as a dynamic interface responding to real-time queries and binding rights and Objects together according to information available at the time of the query.

Example:

Consider the case where bespoke product configurations can be generated dynamically by providing customers with an electronic interface to select combinations from component-based product sets. Combining that with a choice of service levels will lead to a very large number of individual contracts. Building contracts using the relational model (i.e., identifying both content Objects and the expressions of Rights and Obligations governing Interactions over those objects) is preferable to writing bespoke expressions for each product and service-level configuration.

3.2.5.2 Applying one Rights and Obligations Expression to many Digital Objects of the same Type

Requirement:

It must be possible to construct a single rights expression to apply to a class of Digital Objects, without having to copy the rights expression into the metadata for each of the instances. The Language must therefore support class-definitions and mechanisms for identifying to which classes Digital Objects belong.

Note:

There are often too many instances of data items to expect rights to be set manually on a per-item basis. For efficiency reasons, one would not wish to replicate the same rights expressions to each instance. Classes need not be mutually exclusive; i.e., a Digital Object may belong to many different classes. The class might be defined according to attributes of the data, the way in which it is delivered (the medium, the distributor, etc), or via any other feasible mechanism.

Example:

- Rights expression W applies to all data of type “news”.
- Rights expression X applies to all Digital Objects of type “analytic”.
- Rights expression Y applies to all Digital Objects delivered via the pipeline P.
- Rights expression Z applies to all Digital Objects supplied by distributor D
- The class of data labelled “News pictures” may be given distribution rights that need to be modified according to the type of news covered. For example, one distributor may be licensed to distribute only pictures whose topic is fashion. However, geographical limitations may also apply for non-commercial reasons, such as the sensitivities of the recipients. Modern (especially European) fashion photographs may contain displays of nudity, which are unacceptable in certain countries. Therefore, it would be useful to specify in the Rights Language, the restrictions on distribution of such images.

3.2.5.3 Content-Based Matching of Rights and Obligations to Digital Object Instances

Requirement:

It must be possible to express class rights in terms of arbitrary predicate expressions over the values of content attributes.

Note:

Requirement 3.2.5.2 talks about classes of data according to type classifications. A separate requirement has been stated because, typically, type classifications carry the implication of equality matching (e.g., apply rule X *where type = “foo”*). The present requirement is a refinement of the first to allow matching of rights to Digital Object instances according to arbitrary matching expressions predicated on attributes of the instance.

A further refinement of this requirement is stated in Requirement 3.2.5.7, which raises the issue of indirect or fuzzy matching.

Example:

A recipient has the right to access stock prices whose creation date is older than 15 minutes and less than three years. This simultaneously places an embargo on non-delayed real-time prices and archive data of greater than three years.

3.2.5.4 Independence of Content and Delivery Channel where Appropriate

Requirement:

The Language must allow rules to be defined around data classes and content attributes in such a manner that, where appropriate, the rules can be made independent of the channel or mode of delivery of the content.

Note:

This requirement does not preclude the expression of rules in terms of delivery channels where that is deemed necessary. These are two complementary requirements for application in different circumstances.

Example:

A rule may define how a category of "news" data emanating from a particular source may be handled downstream. The rule does not care how the data is delivered, but expresses the requirement for all "news" data of the stated category to be so managed. Therefore, if a distributor obtains the news data from a variety of sources, then all instances are governed by the rule. (NB: It is acknowledged that the behaviour described in this example is not always desirable. Counter-examples exist such that the delivery channel will make a difference to the set of rules applied to the data; especially if the channel may have implications on the integrity or content of the data. This requirement merely asks that both modes of expression are possible.)

3.2.5.5 Default Rights and Obligations when no Expressions have been Defined

Requirement:

Default semantics are required in the case when no Rights and Obligations expressions are currently applicable to a Digital Object. A mechanism to over-ride the default must be made available.

Note:

There are likely to be problems with this requirement with respect to how rights enforcement engines choose to implement default mechanisms. It is safest always to include explicit rules (i.e., not to rely on defaults) even if they simply point to a standard set of terms and conditions.

Example:

It is likely that the default will be defined in the Language to be no access granted.

3.2.5.6 Determining which Rights and Obligations Apply when Multiple Expressions are in Conflict

Requirement:

Default semantics are required in the case when Rights and Obligations expressions are in conflict. This is analogous to the case where conflicts arise due to rights transfer (see Section 3.2.2) but may arise due to multiple expressions being valid simultaneously in the global domain. Alternatively, expressions may have internal conflict. In those cases, the Language must state default semantics for conflict resolution. A mechanism for globally over-riding the default conflict-resolution scheme must also be included in the Language.

Note:

There are many ways in which conflicting expressions may be applied to the same Digital Object. For example, an explicit object ID may be referenced by two different rights expressions. Alternatively, the classification mechanism described in Requirement 3.2.5.2 may lead to expressions written for different object classes applying to a single Digital Object (if that object belongs to more than one class). It may be sensible to treat all conflict equally, irrespective of the manner in which the conflict arises. However, it could probably be argued that some precedence mechanism should be defined. As ever, it would be most appropriate for a default resolution scheme to be defined within the Language and a mechanism for specifying alternative schemes made available. Clearly, one may extend the discussion onto how to resolve conflicting resolution schemes, and the problem of how to avoid that issue is left for further discussion.

Example:

- For financial Obligations, two or more prices might be specified for the same Agent/Object Interaction—one policy might be to enforce the cheapest price.

- One access rule might ban an Interaction on the grounds of group membership. Another (independent) rule might grant access to an individual who happens to belong to the group. If it is possible to determine which is the least general binding (in this case, direct identification of the individual) then it might be sensible to adopt a policy of higher preference being awarded to the least general binding (i.e., access is granted to the individual, despite group membership).

3.2.5.7 Implicit or Fuzzy Identification of Digital Objects

Requirement:

It must be possible in the Language to bind rules to Digital Objects according to fuzzy matching criteria such as “looks like” (with respect to image data), “sounds like” (with respect to audio data), etc. The Standard Prelude must contain an initial set of definitions and enumerations of such criteria.

Note:

This requirement is related to Requirement 3.2.5.3 which discusses content-based matching. The main difference is that content-based matching implies a tight binding between the matching expression and the content whereas implicit identification implies a looser, more fuzzy match that may be subject to the interpretation of technology implementations.

Example:

A Trade Mark is an important legal device for protecting Intellectual Property. A Logo Mark is a particular type of Trade Mark implemented by a graphical artwork. The artwork itself is the subject of copyright, but the Trade Mark carries a different significance. Well known examples include:

- the man made of tyres, marking products from the Michelin company,
- the Sphere device marking products from the Reuters company.

Trade Marks are registered within geographic domains (usually with a national registration authority) and apply within particular vertical commercial categories. In trying to express rules about the permitted use of a Trade Mark in the Language, one would wish to make the rules apply to images that *resemble* a sample artwork. The artwork does not need to be copied verbatim for a Trade Mark infringement to have occurred. While a completely new logo artwork that resembles the original would not infringe the copyright of the original artist, its application could be an infringement of Trade Mark rules. It would therefore not be sufficient for the matching criterion to be an exact copy of the sample file.

3.2.5.8 Unlimited Object Identification Schemes

Requirement:

Provision must be made for Objects to be identified via any suitable identification scheme. The corollary is that the Language must not assume any one preferred identification scheme.

Note:

For any identification scheme one cares to name, there will always be cases where the scheme will not extend to the required identification model (e.g., see the fuzzy matching of Requirement 3.2.5.7). Moreover, if centralised repositories or registries are required in order to service Object identification, one may wish to be able to choose a management scheme which best suits the application from both technical and business perspectives. A single identification scheme would imply a single registration mechanism thus precluding that choice. Another constraint might be on the size of identifier in cases where data efficiency is a premium consideration. A generalised, mandated identification mechanism is likely to compromise efficiency considerations.

Example:

Without wishing to prescribe solutions, the following is an example of how a generalised ID scheme might look:

```
<id scheme="foo">
    Identifier syntax
</id>
```

(with appropriate namespace control over the scope of the scheme name, *foo*).

3.2.6 Matching Rights to Contexts

For the purposes of this section, the term "Context" is used to indicate any situation to which Rights and Obligations may apply. The ultimate consumer may be human or machine, an individual or collective, an application, or any other Agent over which Rights and Obligations apply. In this section, "Consumption" is taken to mean any access to a Digital Object, including handling the Object for the purposes of (re)distribution. To best understanding the complexities, it is useful to build an object model of consumers and the Contexts in which they operate. The concept of an object model is enlarged upon in Section 4.2.

3.2.6.1 Predication via Roles

Requirement:

Role-based identification of Agents is required.

Note:

It is expected that some form of authorisation of the use of the role will be possible, via mechanisms such as PKI signatures. The following variations of the use of roles should be considered:

- Delegated administration (i.e., one person acting on behalf of other users at their organisation)
- Management of groups of users (who all have the same base characteristics)
- Role based activities (where users are known in the context of a role rather than an individual identity).

Note that role-based identification is related to the more general case of context-based predication of rules (see Requirement 3.2.6.2). However, a role is usually formally defined. Consider for example, a private investor. This is not strictly a role because it is not formally assigned and authenticated. However, a private (as opposed to an institutional) investor may claim private investor status on a registration form. In this case, private investor status fits more closely with the generalised notion of an attribute of the context in which the investor operates.

Example:

Example roles, which may be filled by various staff (note that the same staff member may fulfil more than one roles, probably at different times):

- System Administrator
- Customer Services Officer
- Human Resources Manager
- Trader
- Buyer
- Sales Representative
- IT Manager

3.2.6.2 Predication via Attributes of the Context

Requirement:

Provision must be made in the Language for the formation of rights expressions in terms of attributes of Agents (human individual, role, machine, application, etc) and the Context in which the Interaction between Agents and Objects is taking place.

Note:

This is identification according to "something you are", refined by the situation in which Interaction between an Agent and an Object is taking place. It is a generalisation that subsumes predication of rights according to the following:

- the identity of the Agent
- any groups to which the Agent may belong

Example:

- A trader, working for an investment bank, may be making access to market prices from her home PC. Authentication will be undertaken to determine the identity of the trader. Restrictions may apply because the trader is making access across dial-up networks rather than from the bank's intranet. Further restrictions may apply on access according to the declarations by the trader that the access is being made for personal rather than business use (if that mode of access is allowed (i) by the bank and (ii) by the data provider).

- The type of device and network connection may determine access rules. For example, delivery of data to a mobile telephone or PDA may be more restricted than delivery to a desktop PC, for the same Agent.

3.2.6.3 Predication via Possession of a Token

Requirement:

Provision must be made in the Language for the formation of rights expressions in terms of items in the possession of a Token (or set of Tokens).

Note:

This is identification according to “something you have”.

Example:

- A membership token,
- a discount token,
- a certificate of ownership.

3.2.6.4 Arbitrary Predicate Expressions

Requirement:

Data-level permissioning needs sometimes to be achieved via reference to arbitrarily complex predicate expressions. Provision must be made in the Language to accommodate this kind of expression syntax.

Note:

It is likely that the semantics of predicate functions will be defined in local data dictionaries.

Example:

For real-time stock-market data, a data element can be consumed if it maps to a permissioning entity associated with both:

1. a Reuters product purchased by the consumer and
2. a fee paid to the exchange from which the data originated.

For the permissioning entity, *E*, Reuters product code, *P*, (which maps to a set of *E*s) and stock exchange, *X*, (which also maps to a set of *E*s), the permissioning expression on the consumer, *C*, might look something like:

$$C.\text{CAN\_CONSUME}(E) \Leftarrow \exists(P,X) \mid E \in P \cap E \in X \cap C.\text{HAS\_TOKENS}(P,X).$$

In the above expression, the function HAS_TOKENS checks to see whether the consumer is in possession of tokens indicating pre-purchase of subscriptions to products and exchanges. The expression is sufficiently flexible to allow for *E* to be present in more than one product and more than one exchange.

3.2.6.5 Rules-Based Predicate Expressions

Requirement:

For full-flexibility, it must be possible to specify alternative predicate expressions as per a rule-based system.

Note:

Expressions should be compatible with both forward-chaining and backward-chaining rule engines. For example, one might imagine a system that displays a catalogue of all the content to which a consumer has access. This would be best implemented using a forward-chaining style of rule evaluation (especially where the users’ attributes are allowed to change dynamically; for example, by the acquisition of tokens via e-commerce purchases). Where a rights expression is merely to be checked for validity against user attributes, a backward chaining rules implementation would suffice within the rights enforcement engine.

Example:

View, Print, Store ⇐ HAS_TOKEN(gold_service)

View, Print ⇐ HAS_TOKEN(silver_service) ∩ PAYS($2)

View, Print ⇐ HAS_TOKEN(bronze_service) ∩ PAYS($5)

View ⇐ HAS_TOKEN(bronze_service) ∩ COMPLETES(questionnaire(x)) ∩ PAYS($2)

3.2.7 Location, Form, and Access Control of Data Dictionaries

3.2.7.1 Namespaces

Requirement:

It must be possible to identify data dictionaries and libraries of enumerations that apply to a particular rights expression according to a namespace definition similar to that recommended by W3C for XML [7]. Combinations of namespaces must be supported in hierarchical fashion (i.e., definitions from lower-levels of the hierarchy take preference over those from higher in the hierarchy).

Note:

A single framework of enumerations may not suit all. However, for transparency of operation, readers of rights expressions must be able to make reference to the relevant data dictionaries. The namespace idea has worked well for the XML community where this is an established solution to the same problem with respect to XML schema definitions. Namespaces define the scope for terms. The same terms may be defined differently within different namespaces; there is no conflict if namespace declarations are used correctly.

Example:

The URIs http://www.reuters.com/drm/, http://www.reuters.com/drm/video/, and http://www.reuters.com/drm/trading/ each represent notional namespace identifiers. In keeping with the W3C XML namespace recommendation [7], the URIs do not necessarily have to be interpretable as URLs that resolve to particular documents or services. However, the three namespaces might define, respectively, Reuters-wide Data Dictionary definitions, definitions defined specifically for the purpose of Reuters video feeds, and the Data Dictionary terms of Reuters trading solutions.

3.2.7.2 Local Data Dictionary Definitions to Augment and Override Global Definitions

Requirement:

A hierarchy of Data Dictionary terms must be supported such that local definitions may be referenced within closed communities. Local definitions should augment those in the global dictionaries in the most part and, where conflict occurs, over-ride the global definitions. Default augmentation and over-ride semantics are required. It should also be possible to specify alternative semantics in the Language itself.

Note:

See also the requirements in Section 3.1.1 discussing extensibility and management of local definitions.

Example:

- The global data dictionary may specify a particular set of definitions for the term PLAY. However, a new data type defined by an organisation brings new semantics to the term PLAY. The organisation chooses to extend the global definition with its own Data Dictionary terms.
- The global data dictionary may specify a particular set of definitions for the term EXECUTE. The global definition turns out to be incompatible with a particular organisation's needs, so that organisation chooses, within its own namespace, to entirely override the accepted definitions. It might be reasonable to explicitly *undefine* the standard definitions.

3.2.7.3 Data Dictionary Definitions from any Source

Requirement:

The standard must make no assumption about the source of Data Dictionary definitions. Dynamic sources such as LDAP directories and relational databases must have equal standing to the standard libraries and data dictionaries published with the standard.

Note:

This requirement imposes the need for flexibility in terms of the manner in which sources are identified. An assumption that there exists fixed identification syntax is flawed.

Example:

A directory maintained by a company may contain a customer address book arranged according to an internal coding scheme and hierarchy. It should be possible to take data from that book and include it within rights specifications. A good example is that of contributors of data into Reuters systems being able to say which customers are and are not allowed to access the contributed content. Entities named in the address book (at various levels within a hierarchical structure) need to be made available to the contributor in order for them to construct rights expressions for inclusion within larger expressions formed by Reuters when the contributed data is aggregated into a larger service.

3.2.7.4 Integration with External Data Dictionaries

Requirement:

Issues of addressing data sources are to be covered by the standard such that rights expressions can make use of all sources of data definitions.

Note:

This is related to the requirement for namespace-support.

Example:

- An internal LDAP directory containing customer lists may feed into rights expressions.
- Data definitions may flow from a dynamic CORBA, DCOM, or Web Service interface.
- Definitions may be obtained dynamically from a relational (or other) database.

3.2.7.5 Dynamic Data Dictionary Definitions

Requirement:

Where Data Dictionary terms are taken from dynamic sources, such as LDAP directories and relational databases, it must be possible to specify terms in query form for dynamic resolution.

Note:

This requirement implies dynamic rule resolution such that the outcome may differ from one resolution to the next depending on the values returned each time from the dynamic source.

Example:

Rule: allow all operations listed in Directory *Y* against the entry for Agent *X*.

3.2.7.6 Access Control over Data Dictionaries

Requirement:

Commercial process may require that certain Data Dictionary terms be protected by access controls. Therefore, the Data Dictionary itself must be considered a Digital Object and subject to Rights and Obligations constraints.

Note:

Unambiguous identification of Data Dictionaries (for the purpose of defining access controls over them) is assumed possible. See also requirements of privacy and confidentiality in sections 3.5.7 and 3.5.8, describing specifications by downstream Agents of rules to govern the manner in which data about them can be shared.

Example:

Where content is shared between parties in a managed peer-to-peer environment, enumerated lists of named entities, employees, and roles may be key to assigning distribution rights between peers. However, to protect the interests of participants, access should be restricted to the level of information required to specify Rights and Obligations over content, and no more. For example, it would not be acceptable to expose the employee lists of competing organisations to each other via Data Dictionary lookup. Those organisations may, however, wish to specify that access to their content is barred to certain competitor organisations. Therefore, it may be sufficient to expose the list of organisation names with no further hierarchical detail. Other examples may require the level of data exposure to drill down to the level of department names.

3.2.8 Management of Issued Rights and Obligations

3.2.8.1 Lifetime Constraints for Rights and Obligations

Requirement:

By default all Rights and Obligations expressions must persist indefinitely. However, it must be possible for Rights and Obligations to be given bounded lifetimes, where required, beyond which the rules are no longer applicable. Such specifications should be possible at the macro level such that they apply to all Rights and Obligations clauses contained within a set.

Note:

Where Rights and Obligations time out, the question of default semantics arises in terms of which Rights and Obligations are now valid. Clearly, if alternative expressions have been stated and are still current, then those would apply. If no Rights and Obligations expressions are now current then the situation is exactly that addressed by Requirement 3.2.5.5. See also the more general case described in Requirement 3.2.8.7.

Example:

This enables, for example, short-term “special offers” to be created.

3.2.8.2 Revocation of Issued Rights and Obligations

Requirement:

Support should be given for mechanisms by which Rights and Obligations may be revoked during their lifetime.

Note:

This is likely to require methods for identifying specific individual or collections of Rights and Obligations. It may also require support in terms of enforcement protocols in the same manner that the OCSP (Online Certificate Status Protocol) is employed to determine, dynamically, whether a public key certificate has been revoked. There are clearly security implications to be addressed with respect to who is allowed to revoke Rights and Obligations.

Example:

Rights of access to a Digital Object are granted to an individual following an e-commerce transaction. The transaction later turns out to be fraudulent and the rights of access are revoked.

3.2.8.3 Update to Issued Rights and Obligations

Requirement:

Support must be given for mechanisms by which Rights and Obligations may be updated during their lifetime.

Note:

This may be related to the processes for revocation (described above) and renewal (described below). Note that changes to Rights and Obligations may be made after data has already been received if the separation between rights/obligations and data content is maintained as per the requirements in Section 3.2.3. There are clearly security implications to be addressed with respect to who is allowed to update Rights and Obligations.

Example:

A regulatory change alters the reporting rules for reporting of financial transactions. The issued Rights and Obligations do not take the new rules into account. An update is effected so that all existing rules are changed to reflect the new regulation.

3.2.8.4 Renewal of Issued Rights and Obligations

Requirement:

Support must be given for mechanisms by which Rights and Obligations may be renewed once their lifetime has expired (or have been otherwise revoked).

Note:

This may be related to the processes for revocation and update described above. Renewal might be requested by an arbitrary Agent or might be initiated by the publisher; both models should be supported. There are clearly security implications to be addressed with respect to who is allowed to renew Rights and Obligations.

Example:

A business model is created in which short term introductory offers are issued to prospective customers. A temporary set of Rights and Obligations is created to define the bounds of the offer. The Rights and Obligations will, typically, be time-bounded (i.e., the trial period is limited). However, under certain circumstances it is useful to renew a trial offer. Rather than re-issue a whole new set of Rights and Obligations, it might be more convenient simply to renew those already in existence (they may have been crafted especially for the Agent involved, for example).

3.2.8.5 Conditional Update/Refresh of Issued Rights and Obligations

Requirement:

The Language must provide for cases where Rights and Obligations require update or renegotiation when specified trigger conditions are met.

Note:

This requirement takes account of the fact that not all scenarios can be defined up-front, when an original set of rules is defined. Therefore, rather than banning particular applications of Digital Objects entirely, which is an inflexible approach, it is better if certain conditions can be specified as triggers to force an update of the rules currently in place. If, during the intervening period, new rules have been constructed to cope with the new scenario then they can be applied immediately. Otherwise, an out of band process may have to be initiated to cope with the new request.

Example:

- If content is being aggregated into a larger work then existing Rights and Obligations associated with stand-alone content may no longer apply and may need to be redefined.
- Periodic refresh of Rights and Obligations may be required. This may be time-based or usage based. The refresh mechanism would contain sufficient information to make an on-line access, for example, to an automated refresh service.

3.2.8.6 Expression Validation

Requirement:

The Language must provide a means for indicating that a Rights and Obligations expression should be validated by a dynamic (online) mechanism.

Note:

This is equivalent to OCSP in the PKI world for checking validity of certificates at each application.

Example:

Where no mechanism exists for pushing updates, revocations, renewals, etc, to Agents, a pull-mechanism may be preferable to determine that the rule-sets being applied are current. For example, the publisher might supply a URL as part of the Rights and Obligations expression to instruct an enforcement engine to test for changes before applying the expression.

3.2.8.7 Rule Expiration Due to non-Temporal Constraints

Requirement:

In addition to temporal constraints on the lifetime of rights and obligations expressions, it is necessary to allow for arbitrary trigger conditions to expire an expression.

Note:

This is related to Requirement 3.2.8.1, but is the more general case for which, feasibly, any condition may cause rights to expire permanently. Note that this is different from simply including the negative of the expiration condition in the expression itself. That approach would allow the expression to come alive if the expiration condition is no longer met after a period of time. Expiration is permanent, regardless of the ongoing status of the condition that caused expiration to occur.

Example:

Rights are conferred upon a company up until the point that it infringed a set of conditions laid down in a licence. Once infringement occurs, the expression will no longer be valid and the contract must be renegotiated. It is not sufficient for the company merely to stop infringing in order to continue to apply the original licence.

3.2.9 Fail-Over and Behaviour Modification

If rights of access are not granted, it is often useful to suggest an alternative course of action rather than simply deny access.

3.2.9.1 Obtaining Rights

Requirement:

Provision must be made in the Language to define how rights may be obtained through such mechanisms as licence purchase via an e-commerce site. The provision should also allow for the context to determine the location of the source of rights and the terms and conditions of supply.

Note:

The Context in which failure occurred might also determine other features such as the natural language used in to communicate error notifications, invitations to purchase rights, and other such messages. This requirement may usefully be combined with requirements in Section 3.3.3 regarding the acknowledgement of the source of the data and of ownership information for intellectual property.

Example:

A metadata field might be included to allow a URL to be specified for use by a rights enforcement system to direct the consumer to an appropriate Web site if an attempted operation was blocked.

3.2.9.2 Alternative Data

Requirement:

If a rights set denies access to one class of data, the publisher may wish to specify another class of data to which the rights apply. Rendering tools, or rights enforcement engines may be able to use the information to source and supply alternative data.

Note:

This is not the same as bundling several alternative data items in a package and granting access to the most appropriate item. Rather, the requirement is about redirecting to alternative data from source. See also Requirement 3.2.9.3, which discusses behavioural modification rather than alternative data sources.

Example:

A stock-quote is requested by “Ticker Symbol” (a common method for identifying financial instruments) from a real-time interface. However, the Agent has access only to delayed data for that instrument but, instead of refusing the supply the information, the interface redirects the Agent to a source of delayed quotes. The redirection mechanism is almost certainly going to be implemented in a manner that is transparent to the Agent (e.g., in a manner such as HTTP redirect). In that case, the Agent may not be aware that substitute data has been supplied. Clearly, the rules governing this example could have been hard-wired into the data delivery mechanism. It is preferable, however, for the Rights and Obligations Language to be able to code the rules for implementation by a generic (i.e., context-free) data processing engine.

3.2.9.3 Behavioural Modification

Requirement:

It must be possible to specify in the Language alternative behaviours that may be required of Agent Interactions with Digital Objects according to the Context of the Interaction. Transition rules from one behaviour to the next themselves may be predicated upon the Context of the Interaction (i.e., multiple paths should be specifiable with the actual transition path determined dynamically).

Note:

This is richer than expression syntax for describing whether or not a particular Interaction is valid (i.e., where an engine interpreting the rules would return a straight “Yes” or “No” response to a query). The requirement is similar to that of Requirement 3.2.9.2 where rules may specify alternative data. However, this requirement addresses the need for alternative actions allowed on the same data rather than alternative data (which may not be available). By allowing alternative behaviours to be specified in the Language, consistent interpretations of rule-sets can be achieved via different enforcement engines. Without behavioural modification statements, it is left to the implementation of the enforcement engine to determine whether an alternative Interaction between Agent and Digital Object would be sensible, according to the allowed Interactions within the Rights Expression.

Note that information flows become complicated by this requirement because a rendering engine may receive richer responses that simply "Yes" or "No" to a request to perform an action. However, the vocabulary used to specify alternative behaviours back to the rendering engine will be the same as that used to phrase Rights Expressions and Rights queries, so no additional vocabularies need be defined to satisfy this requirement.

Example:

Rules are included within a Rights and Obligations expression governing the rendering of a television film Object in various modes, including high-resolution video, low-resolution video, or audio track only. An Agent attempts to render the Object in high-resolution video mode, but is blocked because her current Context fails to meet the conditions of the Rights expression. However, the behavioural modification syntax within the Language has been used to determine a progression path between alternative behaviours. Accordingly, the rights enforcement engine is directed next to render low-resolution video from the same data Object. The original request is re-formulated in terms of the low-resolution video rendering and, if the Context allows, the rendering engine will proceed to render the Object in low-resolution format. If the Context is such that low-resolution video is also blocked then the process is repeated (according to the specified progression path between behaviours) to try to render the audio track. Only when no further behavioural modifications are present will a "No" answer be returned.

3.2.10 Privacy of Terms Expressed in the Language and Data Dictionary

This section describes privacy issues as they relate to the structure of rules defined using the Rights Expression Language and Data Dictionary. These are essentially requirements for *meta-rules*. See also Section 3.5.6.1, which deals with requirements for expressing, within the Language, rules about the privacy of *other* entities.

3.2.10.1 Keeping Details of Rights and Obligations Expressions Private

Requirement:

The Language must provide a straightforward mechanism for ensuring that the details of Rights and Obligations expressions are private and exclusive to the subjects of those expressions.

Note:

Requirement 3.2.4.3 specifies that expressions of Rights and Obligations may themselves be subject to rules governing access. The present requirement can be satisfied by ensuring that all detailed terms of Rights and Obligations are protected by separately stated visibility rules. It would be better, however, if the structure of the Expression Language were such that the terms identifying the subject of the expression were clearly separable from the detailed terms and conditions applying to the subject.

Example:

An expression may contain numerous clauses applying to different Agents within a complex value-chain. The contract specification is between the publisher and the individuals in the value-chain (who may themselves also add contractual terms). For example, where a three-way commercial relationship exists between a publisher, a channel partner, and a consumer, the publisher may have direct relationships with both the channel partner and the consumer. If the Rights and Obligations of those Agents were expressed within a single document, it might be considered a breach of confidentiality if the consumer had access to the rules pertaining to the channel partner, and vice versa.

3.2.10.2 Obligations Pertaining to the Privacy of Expressions

Requirement:

The Language must provide a straightforward mechanism for defining the Obligations incumbent upon value chain participants with respect to them maintaining the privacy of terms and conditions.

Note:

Requirement 3.2.4.3 specifies that expressions of Rights and Obligations may themselves be subject to rules defining the obligations associated with access. The present requirement could be satisfied by ensuring that access to detailed terms of Rights and Obligations is subject to separately stated obligations. It would be better, however, if the structure of the expression Language were such that *meta-rules* describing Obligations pertaining to the privacy of the Rights expressions could be defined in-line.

Example:

A channel partner may be responsible for passing on Rights and Obligations to a downstream customer. Assuming Requirement 3.2.10.1 is addressed then the channel partner will not be able to gain access to the rules pertaining to the downstream customer. However, there may be an Obligation specified on the channel partner regarding, say, the manner in which the rules expressions are stored locally and delivered to the end customer such that there is no chance of the customer's privacy being compromised by another party.

3.2.11 Expression Evaluation

3.2.11.1 Expression Evaluation Services

Requirement:

The Language must make it possible for specific Expression Evaluation Services to be nominated for the evaluation of Rights and Obligations expressions.

Note:

It may be possible to enforce the rule of which service is used to evaluate an expression by utilising asymmetric (public key) encryption technology. A service might publish a public key with which a publisher encrypts the body of a set of rules expressed in the Language. Only the specified service has the private key required the make access to the rules of the expression. Requirements in Section 3.2.10 describe limits on how the details of Rights and Obligations expression might be made available to Agents. The Service would be subject to rules under those requirements.

Example:

A nominated Web service is identified as the sole trusted entity for evaluating Rights and Obligations expressions. Enforcement/rendering engines are obliged to issue queries against expressions to the service. Queries may result in "Yes/No" answers or more complex behaviour-transforming responses (see Requirement 3.2.9.3).

3.2.11.2 Stateful versus Stateless Expressions

Requirement:

It must be possible to define stateful as well as stateless Rights and Obligations expressions.

Note:

The implementation of enforcement engines and expression evaluators are bound to maintain state where necessary.

Example:

Rights and Obligations are predicated on the number and type of previous Interactions of an Agent with an Object.

3.3 Requirements for Rights and Obligations Definitions

In this section, requirements are given for specific examples of Rights and Obligations to be expressed in the Language of the standard. The examples are based on concrete business requirements and it will often be possible (and better) to satisfy a requirement in more general terms than listed below.

3.3.1 Operational Specifications

3.3.1.1 Quality of Service

Requirement:

The standard must allow the definition of minimum levels of Quality of Service to be met by downstream distributors of content.

Note:

Quality of Service constraints might be applied to contractual and non-contractual partners alike (see also the related requirements pertaining to trust (such as confidentiality requirements) in section 3.4.3.1). For example, an authorised distributor of content will be contracted to supply services while an edge-server or network cache might not be under contract to the content owner

Example:

- A distributor must deliver content to downstream customers within two seconds of receipt.
- A maximum planned outage of two hours per year is allowed.

3.3.1.2 Trust-levels

Requirement:

It must be possible to specify a minimum level of trust to be established between a distributor of content and a consumer before the distributor is permitted to deliver the content to the consumer.

Note:

Trust levels are typically associated with well-defined business practices such as validation and verification processes employed during customer registration. Usually, trust levels will defined on a per-organisation basis. There is presently no notion of globally defined trust-levels.

Example:

- Content may only be access via a specified trusted application
- Authentication of the consumer must be made via an approved method (e.g., using public key certification with a specified root of trust)
- Only digital certificates conforming to a specified trust-level are acceptable for authentication and authorisation purposes

3.3.1.3 Application of Client Fingerprints to Content

Requirement:

There is a requirement to insist that data delivered to a recipient be marked in some manner to identify the recipient. It must be possible for fingerprints to be added at various stages in the distribution of data and for both the fingerprints of distributors and end-consumers to be required.

Note:

If the data finds its way to an illicit destination, the identification marks (or fingerprints) can be used later in legal proceedings to implicate the legitimate recipient who passed the Digital Object to the illicit destination. The ability to fingerprint is dependent on the type of data being delivered. The application of fingerprints may be performed by distributors, once the end-consumer is known, or by some enforcement device at the consumer site. The requirement for Language specification is agnostic of the implementation.

Example:

A distributor of news pictures is responsible for delivery to end-consumers. The distributor takes a wholesale feed from the publisher. The publisher is concerned that image data should be traceable back to the customers of the publisher in case illicit copies of the images begin to appear on unlicensed Internet sites. The publisher insists, therefore, that the distributor adds watermarks to images sent to customers such that the watermark contains customer-identification information. The information need not be accessible by the publisher, but the distributor must be able to provide audit information, as necessary, to identify the source of leaked images if licence infringements occur.

3.3.1.4 Caching and other Network Operations

Requirement:

Limitations on intermediate caching servers must be specifiable in the Language. Many types of caching need to be addressed, including caches within client organisations, generic network caches, and edge-servers. This requirement must generalise to any kind of network service, including those not yet defined.

Note:

This requirement is directed at network-level operations and devices. It is not intended to indicate modes of usage on behalf of recipients of data. See also Section 3.3.8 for usage rights, which includes requirements for expressions dependent upon the way in which Agents handle Objects (e.g., hold them in local databases). This mode of permissioning is particularly relevant to publishers who wish to retain control over their data while it is in transit across public networks.

Example:

A publisher responds to a request from a client for a valuable research report. An electronic copy of the report is sent across a public network. En route, the Object passes through numerous network operators' domains (e.g., the publisher's ISP, then into the Internet "cloud" wherein it may pass through any number of domains of control, and finally via the client's ISP to the client herself). At any stage between publisher and client, a network device may cache the Object with the intention of improving the efficiency of the network. If the publisher objects to such caching then a simple rule attached to the Object denying caching Rights should instruct caches to ignore the Object.

Clearly this example raises questions. Firstly, identification of caching devices is "fuzzy". A device must know that it is a cache in the sense indicated within the rules of the Object. It must also know that it is supposed to read the rules (but we assume that network standards will develop to the point that Rights expressions on network-level devices are a well-understood phenomenon). Ultimately (assuming that the technology approach fails to protect the IPR of the publisher), the point of expressing such rules is to provide a basis for litigation, should that prove necessary. In that case, a judge will determine whether or not the device should have obeyed the rules. One would hope that well-behaved devices would avoid the need for legal recourse (N.B., this hope is not a weakness of the requirement but, rather, a commentary on how the industry is most likely to move forward).

Another question is why does the publisher simply not rely on encryption techniques if it wishes to protect its IPR? To answer that, simply return to the premise stated in Section 0, "the essence of Rights and Obligation Management is electronic contract management". The rules expressed in the Language define the contract. Encryption of content is an orthogonal issue. The publisher might choose to use encryption *and* to express rules for caches.

3.3.1.5 Style Guides

Requirement:

It must be possible to insist that content is used in a specific manner, such as by the application of style guides and templates to define "look and feel".

Note:

The style guides themselves would not form part of the Language—they would be defined according to an appropriate style guide standard. However, the mechanism for insisting on the application of a style guide would be part of the Language.

Example:

- An HTML template file (or XML-based style sheet) may be issued into which components should be plugged.
- Machine-readable rules about text-styles (e.g., fonts, spacing, etc) may be defined.

3.3.1.6 Trust Services

Requirement:

It must be possible to specify a particular mechanism for determining whether or not trust criteria have been met.

Note:

It is likely (but not essential) that this will involve invocation of a dynamic trust service that will inspect the context to determine the appropriate trust level.

Example:

A trust service may evaluate the context in which Digital Objects are being handled or consumed according to some defined method (e.g., possession of a public key certificate issued by a notary subject to pre-determined criteria for determining trust levels).

3.3.2 Reporting

3.3.2.1 Usage Reporting

Requirement:

It must be possible to define in the Language the required levels of usage reporting on data supplied downstream within a value-chain. The specification must allow the obligation to be specified on anyone in the value-chain, including distributors, re-sellers, and consumers of content. It must be possible to include constraints such as maximum time lag for delivery of reports and frequency of reporting.

Note:

This is an obligation on distributors and others in return for being allowed to handle content. Privacy rules should be obeyed. Reporting may be at a less granular level than individuals.

Example:

Channel partners and distributors of content are obliged to report on usage of the content by their customers.

3.3.2.2 Financial Reporting

Requirement:

The Language must allow the specification of financial reporting requirements on downstream value chain participants. It must be possible to include constraints such as maximum time lag for delivery of reports and frequency of reporting.

Note:

This is an obligation on distributors and others in return for being allowed to handle content

Example:

A channel partner is obligated to report once per month on royalties owed to a publisher for vending the publisher’s content. The report must be broken down into separate royalty accounts according to specified product categories.

3.3.3 Acknowledgement of Source

3.3.3.1 Branding

Requirement:

The Language must make provision to describe requirements on distributors and other value-chain participant for branding content. Metadata constructs for describing the branding material (e.g., logos etc) are required.

Note:

Branding is not a straightforward case of simply specifying a logo to attach to a document. The specification may need to include details of precisely how the brand is to be represented (e.g., size, position, colours, fonts, duration, prominence, etc). All media types can be branded, so the requirement covers, for example, specifications of how long (i.e., on how many frames) a logo might be displayed on a video and during which part of the video (e.g., within the first N seconds, at the end, or constantly throughout). Given the complexity of branding, the Language should adopt a flexible and extensible approach for rule definitions governing the manner is which brands are represented.

Example:

- A news agency may insist that its name is associated with all of its textual news stories published on Web sites. Additionally, for certain customers, Web pages may be required to include a specified logo image with links back to the agency's own Web page.
- A television news company's logo is required to be displayed for the first five seconds whenever video news footage is used in certain contexts by third parties. Different rules apply according to geographic and other constraints, including special rules for named third parties.

3.3.3.2 General Acknowledgements

Requirement:

The Language must allow for required acknowledgements to be specified as a consequence of using data.

Note:

Metadata fields should be defined as appropriate to handle the descriptive data.

Example:

- The name of the publisher and photographer should be stated whenever an image is printed.
- Acknowledgement of the New York City Mayor's Office is required for films made on location.
- Television programmes syndicated to third parties carry rules about the running of credits at the end of each programme (third parties are not at liberty to cut the credits out to gain extra advertising time, for example).

3.3.3.3 Legal Notices

Requirement:

The Language must make provision to describe requirements on all participants in the value chain (including rendering tools used by the end-consumer) on the placement of legal notices alongside content.

Note:

There may be a general-purpose mechanism that can handle the requirements describing acknowledgements, branding, and legal notices. However, a clear distinction need to be drawn between these in order that the resulting metadata can be processed mechanically in an appropriate manner. The purpose of including the metadata is not simply for human readers, but so that machines can deal with content in an appropriate manner. To that end, vocabularies defining the terms of the expressions in a tightly defined manner would be extremely useful. Those vocabularies probably will not form part of the Language, but would be taken from other sources (e.g., a legal markup language).

Example:

- Copyright statements are a prime example of a legal notice.
- Statements of terms and conditions such as the familiar "This book is sold subject to the condition that it shall not… be lent, re-sold, … in any form of binding or cover other than that in which it is published…"
- General assertions of rights, such as an assertion of the moral right to be named as the author of a work.

3.3.4 Rights and Obligations for Real-Time Data

3.3.4.1 Fairness of Delivery

Requirement:

Provision must be made in the standard so that content originators can specify on their downstream partners the requirement for fair delivery schedules. This is a generic requirement and may be refined for specific cases. The Language must therefore provide extensibility in this area.

Note:

Under fair delivery rules, recipients of data should not be placed at a measurable disadvantage because they received data late due to network effects and distribution mechanisms. Random disadvantage may be allowed (i.e., where, for each separate event, delivery will be unfair; however, averaging over a large number of events, fairness prevails). Therefore, different classes of “fairness” may need to be defined.

Example:

Recipients of real-time stock quotes would be unfairly treated if, for example, a round-robin delivery mechanism were used which always delivered data to customers in the same order. If a channel-partner were delivering this data on Reuters behalf, then it should be possible for Reuters to place the obligation on the distributor that data is delivered fairly.

3.3.4.2 Timeliness

Requirement:

It must be possible to specify in the Language, allowed time-windows for data delivery. The obligation is on downstream distributors and channel partners to uphold the quality associated with the originator’s brand when they are responsible for delivery to the next link in the value chain.

Note:

This is a specific example of a QoS obligation (see Requirement 3.3.1.1). Additional complexity such as averaged timeliness might also be useful (e.g., on average, data must be delivered within specified time constraints).

Example:

A distributor must deliver real-time data to client sites within 2 seconds of receipt.

3.3.4.3 Bandwidth

Requirement:

It must be possible to specify in the Language minimum bandwidth to be operated by downstream distributors and channel partners when they are responsible for downstream real-time data delivery.

Note:

This is a specific example of a QoS obligation (see Requirement 3.3.1.1). Bandwidth specifications are complex and may include (for example) measures of average rates, peak rates, and median rates. Where data is “bursty”, constraints may be specified with respect to the maximum allowed time lag during busy periods.

Example:

The provider of real-time stock-quote information via channel partners may insist that the channel partner support a minimum bandwidth in order that it does not either drop updates or fall behind during peak periods.

3.3.5 Rights and Obligations for a Stream of Digital Objects

3.3.5.1 Stream-level Permissioning

Requirement:

A basic level of permissioning may be specified at the level of the stream. Therefore, provision must be made to identify the stream as a Digital Object in its own right and to link rules of access and distribution to that object.

Note:

Note that the transfer of Rights and Obligations between streams and the objects within the stream is covered by requirements listed in Section 3.2.2

Example:

Access to a stream may be predicated on a subscription fee being paid to a basic service.

3.3.5.2 Content-level Permissioning of Streams of Objects

Requirement:

In addition to permissioning at the granularity of a whole data stream, it must be possible to permission access and distribution to a finer granularity based on the data content flowing within the stream.

Discrete content packets within the stream must therefore be identifiable, and associated with the appropriate rules. The rules will not, however, be packaged with the individual data items. They will be specified at the stream-level, but predicated on attributes of content packets.

Note:

This requirement is related to Requirement 3.2.5.3 and essentially states that content-based matching, as expressed in Requirement 3.2.5.3, be combined with Requirement 3.3.5.1 to define a filter over streamed objects. The filter effectively defines a sub-stream (which is itself a virtual Digital Object) against which rules apply. Note that many virtual streams might be so defined within a physical stream and that the virtual streams may overlap (i.e., one Digital Object may be a component of several virtual streams). This may be a source of rule conflict and will require resolution as per Requirement 3.2.5.6.

Example:

A stream may contain a mix of real-time and delayed stock quotes. Rules of access for real-time quotes within the stream will be different to rules of access for delayed quotes. Therefore, two virtual streams exist within the single stream and attributes of Objects within the stream determine to which virtual stream the Objects belong and, hence, which rule-sets apply. A certain customer may have access to the stream but may be limited to delayed quotes. All real-time quotes must be filtered out by the rights enforcement component of the rendering engine or local data distribution mechanism.

3.3.5.3 Efficiency of Permissioning / Minimising Overheads

Requirement:

Real-time streaming data is often high-bandwidth and subject to advanced compression techniques. The imposition of significant overheads by the addition of access control and other rights management syntax is unwelcome. Consideration must be given to the efficiency of rights expression in this context.

Note:

At one level, the Language design shot not force implementers into inefficient designs. At another level, it might be possible to build support for efficiency into the Language (for example, by allowing stream-based rules to be defined at the stream level and matched to individual data components as per Requirement 3.3.5.2).

Example:

Real-time pricing updates may be in the order of a hundred or so bytes. Even a 10% overhead in additional information would place an unreasonable burden on the systems processing and transmitting the data.

3.3.6 Rights and Obligations for Transactional Data

See also Section 3.3.9 (Managing Communities) for requirements pertaining directly to the members of communities. That section is relevant to transactional data because transactions are often carried out within a community context.

3.3.6.1 Settlements

Requirement:

The Language must make provision for rules specifying when particular actions must be completed.

Note:

The settlement rules might be useful in driving automated systems to ensure that payments are made as late as possible to ensure compliance.

Example:

A typical settlement period for a stock exchange is 5 days after the transaction is struck.

3.3.7 Rights and Obligations for Database or Server Access

While one may consider databases and servers to be centralised resources which already possess permissioning mechanisms and for which access rules can be defined centrally, there remains a general requirement to include the access rules within the scope of the Standard Language. The primary reason is that policies can be defined within the Language and propagated to distributed and heterogeneous servers on the network which then enforce the rights via their individual implementations (i.e., the database or server is effectively implementing a rights enforcement engine).

3.3.7.1 Depth of history

Requirement:

It must be possible to express restrictions on the formation of queries according to the age of data being retrieved from historical data sources.

Note:

This is a more specific example of Row-level permissioning described in Requirement 3.3.7.3.

Example:

- Only the last year's news history may be searched.
- Only prices older than 15 minutes may be retrieved.

3.3.7.2 Limiting Server Load

Requirement:

It must be possible to express restrictions on the execution of queries according to the load imposed on the server (specified according to any useful measure of load).

Note:

This requirement may be more generally applied to any Digital Object performing services on the network.

Example:

- A database query may take no longer than 5 seconds of CPU time to execute.
- No more than 10% of available core memory is allowed for each client request.

3.3.7.3 Row-Level Permissioning

Requirement:

It must be possible to express restrictions on records retrieved from the database according to attributes within the individual records.

Note:

This is similar to Requirement 3.2.5.3 (Content-Based Matching of Rights and Obligations to Digital Object Instances) but, here, the database records need not be treated as independent Digital Objects. The subject of the rule is the database interface.

Example:

In retrieving records from a vehicles database, the Agent has access only to records of cars owned by company X.

3.3.7.4 Column-Level Permissioning

Requirement:

It must be possible to express restrictions on what data elements may be retrieved from the database.

Note:

This is directly analogous to the idea of providing *views* on a relational database.

Example:

A human resources database contains many sensitive fields within employee records. Access to those fields is controlled according to the role of the Agent. Human Resources managers have full access. Line managers have access to salary and employment record details, but cannot see personal information such as notes of confidential discussions held between Human Resources and the employee.

3.3.8 Usage Rights

It is important to note that usage rights apply to all participants in the value chain, not just to end-consumers. For many companies, such as Reuters, customers are rarely end-consumers. For example, news customers tend to be newspapers, magazines, TV stations, radio stations, Web sites, and so on. None of those are "end-consumers" in the B2C sense, but it is just as important to be able to specify the rights of those customers.

3.3.8.1 Delivery Medium

Requirement:

The allowed media of content delivery must be specifiable. Specifications must allow for negative definitions. Obligations and rights may be contingent upon the delivery medium.

Note:

The delivery medium may require a complex taxonomy to be included in the Data Dictionary.

Example:

Current examples include TV, radio, Internet, magazines. However, it is to be expected that those relatively coarse grained categorisations may require further refinement in future. TV may, for example, require sub-division into cable, terrestrial, satellite, etc.

3.3.8.2 Limitations for Purpose of Consumption

Requirement:

It must be possible to contract with consumers of content that the content will be used for specified purposes.

Note:

This is distinct from modes of consumption such as "view", "print", "save". Each of those activities might be performed for the same purpose. Alternatively one might "view" a document with different purposes in mind on each occasion.

Example:

Content might be licensed only for the purpose of providing backing information to the consumer and not for the purpose of trading.

3.3.8.3 Liability Statements

Requirement:

It must be possible within the Language to describe liability incumbent upon value chain participants if they agree to Interact with Digital Objects in a particular fashion.

Note:

Liability may be on any value-chain participant (i.e., the publisher, the consumer, or any other entity).

Example:

A distributor disseminating real-time stock pricing may undertake to ensure that fair-delivery of prices is achieved (i.e., that no one client is systematically disadvantaged by the delivery mechanism). A condition of distributing the data may be that the distributor accepts any liability associated with this undertaking.

3.3.8.4 Aggregation

Requirement:

The standard must include expressions to define terms and conditions (Rights and Obligations) for a third party aggregating content within a larger content offering.

Note:

This includes aggregation both with content:

- solely from the same publisher and
- a mixture of sources.

The rules may be complex and there is a strong requirement here for extensibility.

See also Requirement 3.3.8.13, "Rights of Reference".

Example:

▪ A company's data cannot be combined with that of a competitor.

- The context in which data can be aggregated into a larger composition is governed. The rules specifically ban inclusion into material of specified category types (e.g., clips taken from video footage of a football match are not allowed to be included in sports programmes, but are restricted to scheduled news programmes.).

3.3.8.5 External Context

Requirement:

The standard must include expressions to define valid "real world" contexts in which the content may or may not be used. This includes both complete content items and sub-components of the content (for example, quotations).

Note:

This is distinct from limitations predicated on delivery channel or medium, as described in requirement 3.3.8.1. The enumeration of contexts would be defined within the Data Dictionary, not the standard itself. This is different from Requirement 3.3.8.4 because that limits the kind of material with which Objects may be aggregated whereas the present requirement limits the more general context, which may be a "physical-world" scenario. This kind of constraint may well require legal and business implementations for enforcement rather than technical solutions.

Example:

The use of Reuters content by pornographers would not be sanctioned, regardless of delivery channel or type of document.

3.3.8.6 Alteration of Digital Objects

Requirement:

The Language must make provision for rules defining how Agents may alter Digital Objects subsequent to publication.

Note:

Rules may carry both limitations on alterations and consequences (obligations) of altering the content.

Example:

- Examples of way in which contented may be altered include (but are not limited to):
  - Précis
  - Reformat
  - Resize
  - Clip
  - Stretch
  - Change Font
  - Transform
  - Annotate
- Examples of consequences for altering content may include (but are not limited to):
  - Explicit statement of alteration required to indemnify publisher against errors introduced
  - Links back to the original Digital Object

3.3.8.7 Annotation of Digital Objects

Requirement:

The Language must allow for rules governing the manner in which annotations may be added to an original Object. It must be possible for the rules to describe how annotations are presented and distinguished from the original content.

Note:

Annotation is a special case of alteration, restricted to the addition of annotative material. With uncontrolled annotation, it is possible to circumvent rules about altering content. Consider the example phrase "The President stated that the weapons will be decommissioned." Adding an annotation in a well-defined manner, such as by enclosure within square brackets, is useful for adding additional information: "The President stated that the weapons will be decommissioned [Source: White House press release, May 3]." Allowing annotations to be slotted in arbitrarily may lead to the sense being changed; note the effect of the annotation "not" in the sentence: "The President stated that the weapons will not be decommissioned." The question is what differentiates an annotation from a standard alteration of the content? The annotation rules provide the necessary distinction.

This requirement is related to Requirement 3.2.4.2, which talks about deriving metadata from an Object and its subsequent publication.

Example:

In a particular literary work, annotations may be added only as margin notes or via a mechanism such as standard footnotes. The syntax for annotations must be clearly explained in the pre-amble of the annotated text.

3.3.8.8 Use of Sub-Components of a Digital Object (Internal Context)

Requirement:

It must be possible to express rules about sub-components of a Digital Object using both specific and general terms of reference. The rules may express relationships between components or may simply constrain individual components.

Note:

General terms of reference are useful for describing (for example) rules about

1. generic sub-components of the Digital Object (e.g., *any paragraph* of text)
2. the Digital Object as a whole
3. fractions of the Object

while specific expressions relate to explicitly identified components within the object.

Example:

- No quotations may be taken from this object; the object must be distributed in its entirety.
- Up to 25% of the text of this object may be used for purpose *X*.
- The first paragraph may be extracted, but only in the context of the original title and other specified metadata.
- Paragraph 7 may be extracted for Web publication in its own right, but only if accompanied by the text of Paragraph 5. Neither paragraph may be altered in any way.

3.3.8.9 Digital Object Retention

Requirement:

It must be possible to state rules for how a Digital Object may be retained for future use. The access context may be different for a Digital Object accessed directly from an authorised supply mechanism than it is for the same object accessed from an Agent's local store. The context may alter merely because the Object is retained in a local store or be dependent upon criteria such as the number of accesses previously made to the retained object or the time for which it is retained.

Note:

Both rights to access the object and obligations incumbent upon the Agent may change according to retention policies. Limits may also be placed on the time period for which material may be archived. This leads to a two-tier time frame: the first period—e.g., 72 hours—in which rules for the live-feed apply, and the second period—e.g., 90 days—in which rules for the archived content apply. See also the related policy discussions pertaining to network elements such as caches described in Requirement 3.3.1.4 (caches are not consumers of content but, rather, conduits which may retain the content in transit).

Example:

Video footage supplied on a live news-feed has broadcast rights conferred according to the contract for taking the news feed. However, if a customer records the video footage in a local archive, additional rights may need to be negotiated before it is used again. Moreover, certain video sources may specify that archiving is not permissible.

3.3.8.10 Digital Object Deletion and Destruction

Requirement:

The Language must provide the facility to describe rules about deleting Digital Objects in an unlimited variety of contexts and circumstances. The rules must extend to the manner of destruction of the Object, which will include such requirements as caching a deleted Object within a retrieval area until a condition is met (timeout, flushing of the cache, etc).

Note:

Deletion may be from (amongst other cases):

- a collection of objects,
- a globally accessible storage area,
- a local file system,
- a physical device.

Example:

Restrictions describing who may delete a file inserted into a persistent, distributed, peer-to-peer file storage and publication system such as Publius [6]. The authentication mechanism required to satisfy the delete rule may also need to be specified.

3.3.8.11 Interaction with Digital Objects

Requirement:

The Standard Prelude shall provide a rich set of enumerations (supported where necessary by structure in the Language) for *at least* the following modes of interaction:

- Alter
- Analyse
- Approve
- Destroy
- Execute
- Initiate
- Process
- Render
- Search
- Sign

In addition to enumerations in the Standard Prelude, mechanisms for describing constraints will be included in the Language.

Note:

The above list is by no means exhaustive. More specific requirements on several of the modes of consumption are listed elsewhere. This requirement is intended as a *catchall* to ensure completeness of this requirement specification. Requirement 3.3.8.12 is closely related and describes the handling (rather than direct consumption) of Digital Objects.

See also the requirements on extensibility and localisation of data dictionaries elsewhere in this response for discussions about how such enumerated lists can be managed and extended.

Example:

- Examples of enumerations for “Render” may include (but are not limited to):
  - Display
  - Print
  - Save
  - Play
  - RenderAsHumanSpeech

3.3.8.12 Handling of Digital Objects

Requirement:

The Standard Prelude shall provide a rich set of enumerations (supported where necessary by structure in the Language) for *at least* (but not limited to) the following modes of handling Digital Objects:

- Export
- Destroy

- Lend
- Copy
- ReSell
- GiveAway
- Combine
- Distribute
- Redistribute
- Retain
- Aggregate

In addition to enumerations in the Standard Prelude, mechanisms for describing constraints will be included in the Language.

Note:

The above list is by no means exhaustive. More specific requirements on several of the modes of handling are listed elsewhere. This requirement is intended as a *catchall* to ensure completeness of this requirement specification. Requirement 3.3.8.11 is closely related and describes the direct consumption (rather than mere handling) of Digital Objects.

See also the requirements on extensibility and localisation of data dictionaries elsewhere in this response for discussions about how such enumerated lists can be managed and extended.

Example:

Examples are contained within the text of the requirement.

3.3.8.13 Rights of Reference

Requirement:

It must be possible within the Language to specify rules regarding how content may be referenced. This is different from, but related to, rules about direct aggregation and includes such methods of reference as linking via URLs.

Note:

See also Requirement 3.3.8.4, "Aggregation".

Example:

A Web site is a Digital Object. When utilised by a business partner as a live resource, a publisher may ban deep linking into certain areas of the Web site in order to avoid loss of advertising revenue. The right of reference is granted solely to the publishers home page, or to carefully selected jump-off points within the site that preserve advertising revenue.

3.3.9 Managing Communities

3.3.9.1 Chinese Walls

Requirement:

The Language must provide for the expression of Rights and Obligations describing which parties may converse with each other within a transactional (or other) system, and limitations on the nature of the conversations.

Note:

These rules will be used to reinforce "Chinese Walls" (which may be required by law, especially in transactional environments). Implicit in this requirement is the need for unambiguous identification of conversing parties. The right to take part in a transactional, or other, activity may be predicated on the observance of "Chinese Walls". Therefore, the rule must be expressible in the Language.

Example:

Within a single organisation, different personnel may function in different trading roles (e.g., straight brokerage, acting on behalf of clients, without the broker personally taking a market position, versus market-making activities where the trader takes a position in order to create liquidity in the market). Regulations often require limited conversation between those parties. Similar requirements might be generated outside of the regulatory domain where conflicts of interest are detected.

3.3.10 Contract Management

3.3.10.1 Contract Specification

Requirement:

Provision must be made in the Language for contractual terms to be specified.

Note:

This is an example where the Specification Language is probably outside the scope of the Rights Expression Language. Appropriate contract definition standards should be identified, where they exist. A typical Rights and Obligations specification may require a number of discrete contract specifications given that the contract may need to be as granular as the Rights and Obligations specification itself. Without the ability to componentise the contract expression, it is unlikely that a monolithic contract specification could be integrated well with a monolithic rights expression. No limit should be placed on the form of the contract specification, but some method of identifying the type of contract specification would be useful.

Example:

- Contract specification language X is used to define a contract to apply to access to audio content within a data package by a consumer when the data is downloaded to an MP3 player. The contract syntax might be bounded by tags resembling:

  `<contract type=”structured” language=”X”>`*text of contract*`</contract>`

- A free-text contract specification is used to define a contract to apply to a channel partner distributing the same data package in its entirety. The contract syntax might be bounded by tags resembling:

  `<contract type=”unstructured”>`*text of contract*`</contract>`

3.3.10.2 Workflow for Contract Establishment

Requirement:

Where explicit acknowledgement of contractual terms is required by a publisher before access to an Object is granted, the Language must provide syntax for specifying both the requirement for acknowledgement and the acceptable means by which acknowledgement can be made. The Language must allow for all possible commercial relationships. For example, specifications may form direct contract establishment between a publisher and a consumer, or may be specified in terms of contract details to be passed downstream via a distributor with an acknowledgement required back from the final consumer (either via the distributor or direct to the publisher).

Note:

This requirement describes obligations on both end users and distributors of content. See also Requirement 3.5.4.1, which discusses the construction of audit trails.

Example:

- A publisher requires a distributor to construct a mechanism via which an authenticated consumer can view an electronic “click through” contract and that the consumer can acknowledge the contract by clicking on an appropriately labelled “accept” button. The publisher further requires the distributor to log the identity of the consumer and the date and time at which the accept button was clicked.
- The above example might be further constrained by specifying acceptable authentication mechanisms for consumers or by replacing the click-through aspect of the contract with workflow that requires a public key signature from the consumer to acknowledge the contract.

3.3.10.3 Explicit Acknowledgement of Individual Contractual Terms

Requirement:

The Language must make it possible to list explicit contractual terms (i.e., Rights and Obligations) that each require explicit acknowledgement.

Note:

The acknowledgement will be enforceable as a guarantee on the part of recipients of the terms that they will abide by the terms as stated and that they undertake to perform any obligatory actions according to specified constraints.

Example:

A distributor is required to perform an editorial function on material supplied (such as constructing highlights from a sports fixture) and to pass on the value-added derived work to downstream customers as part of the contract. Moreover, the distributor is bound to a maximum amount of material that may be released to each downstream customer, per fixture, and may not release material beyond a specified time after the original event took place. Three separate acknowledgements are required, the first of which binds the distributor to undertaking work while the remainder simply acknowledge constraints on distribution.

3.3.10.4 Multi-party Contracts

Requirement:

The Language must make provision for multi-party contracts.

Note:

This includes both specifications of rules pertaining to multiple parties and to contractual workflow involving multiple parties.

Example:

Content is vended via a channel-partnership arrangement, such that a portal site provides direct access back to a supplier of services. Customers register with the portal site and there may exist direct relationships between customers and the portal, the portal and the service supplier, and customers and the service supplier. One may imagine complex contractual arrangements describing the separate and joint liabilities of the portal and supplier with respect to the end consumer, and the liabilities of the portal and service supplier to each other. Situations might exist, for example, where the portal acts partly as an agent of the supplier, running first-line customer-support.

3.3.10.5 Persistent Obligation to Seek Permission for Content Use

Requirement:

A special case of contract establishment is one which requires a value-chain participant (Agent) to seek explicit permission for each and every use of a Digital Object. The Language must provide a method for describing this case.

Note:

This is an example of an obligation on an Agent.

Example:

A photographic image may be “purchased” from a news agency for use by a magazine. The contract states that the image may be used once only for a particular publication, but allows for the image data to be stored on the magazine’s database for up to 90 days after it is purchased. However, any and all subsequent uses of that data within the 90-day storage period are subject to permission being granted explicitly by the publisher (and, potentially, payment of an additional fee).

3.3.11 Business Models

The majority of the following requirements are examples of specific payment obligations that need to be accommodated within the framework. The combination of an extensible Language and an extensible Data Dictionary should be able to handle all of the examples by virtue of extensibility. However, a core set of business models should be expressed in the Standard Prelude. It may be the case that an external standard already addresses some or all of the payment mechanisms.

3.3.11.1 Charging Models

Requirement:

The Standard Prelude must contain *at least* (but not limited to) the following simple charging models:

- Free access

- Pay-per-Interaction
- Pay outright for unlimited allowed Interactions
- Pay-per-Interaction up to a limit and then switch to free access for allowed Interactions ("rent-to-own")
- Pay a minimum up-front charge on a subscription-basis and call per-Interaction charges off against the initial charge until it expires either due to timeout or because it is used up. Subsequent Interactions attract separate charges.
- Sponsored charging models (where charges are applied to a different organisational unit than that causing the charge).
- Discounting schemes (e.g., discounting applied once tiered volumes are purchased)

Note:

The models outlined above are far from exhaustive. The list is intended to give a flavour rather than a definition of required charging models. Financial charges applied to Object Interactions are just an example of an Obligation rule. The above may be combined with other functionality such as time-limited rules (e.g., free access for a trial period). Discounts according to Context are also achievable simply by specifying different charging models on a Context-dependent basis—this does not require a separate charging model definition. See also Requirement 3.6.1.2 for a discussion of additional complexities of B2B environments.

Example:

Mobile telephone companies often apply the "free-calls as part of line-rental" model whereby a maximum of, say, £5 of the line-rental is made available to cover calls. Any calls in excess of the initial £5 are chargeable in addition to the line-rental.

3.4 Attributes on which Rights and Obligations are Predicated

This section contains concrete requirements for attributes against which Rights and Obligations may be expressed. Note that many such attributes are covered elsewhere in this response. This section provides a catchall to ensure inclusion of attributes not covered elsewhere.

3.4.1 Temporal

3.4.1.1 Time-Based Embargoes

Requirement:

It is often the case that a publisher releases content to a distributor or aggregator ahead of time in order for the content to be manipulated in some manner. The Language must allow the publisher to define the conditions for releasing the content at specific times and dates.

Note:

Time-based embargoes are limitations on when the content is available for wider distribution. See also requirements in Section 3.5.2 for rules governing trusted time services.

Example:

- In the physical world, books are sent to retailers ahead of release dates for logistical reasons. The retailer is bound to keep the book secret (witness J.K. Rowling's tight control over the release of her "Harry Potter" books). There is no reason to think that similar logistical issues will require electronic copies of content to be released into the systems of digital distributors ahead of time and that the electronic copies will be subject to the same kind of time-based embargoes.
- Regulatory news distribution (i.e., the controlled dissemination of news items about corporate events and changes that may have an effect on stock prices) carries strict rules on the sequence and timing of releases. Increasingly, regulatory authorities are delegating responsibility for the distribution to news industry players rather than handling the distribution themselves. In those cases, both the regulatory authorities and the companies (and their PR agencies) issuing regulatory news stories will impose rules about release schedules.

3.4.1.2 Time of Day

Requirement:

Rights and Obligations may change according to the current time of day. This must be expressible in the Language.

Note:

The Language definition may need to say something about time zones (e.g., all times specified in terms of UTC). Alternatively, the Language will need to accommodate time zone specifications (which are almost certainly addressed in other standards). Times may also be specified in absolute or relative terms or with reference to external definitions (e.g., timeframes managed by external authorities, such as allowed trading hours).

Example:

- The terms and conditions for accessing stock data may depend upon the official opening hours of a stock exchange.
- Rating systems apply differential rates of charge according to times of day (e.g., for making telephone calls or consuming electricity). These are usually driven by the capacity of the system and the times of day when peak loading is expected. Digital equivalents may include peak rates for accessing real-time data streams at certain times of day (e.g., for the first hour after the New York Stock Exchange opens), or global Web site accesses during the overlap between North American and European working days. Many other examples exist.

3.4.2 Geographic

3.4.2.1 Geographic Predicates

Requirement:

Certain rules will require modification within certain geographic domains or may apply only within certain geographic domains. It must be possible to specify predicate expressions in geographic terms for this purpose.

Note:

Standards for naming geographic domains will no doubt be useful. However, it is not sensible to restrict identification to a single standard (e.g., ISO-three letter country classifications) because such standards rarely (if ever) encompass all domain descriptions that will be needed. Consider, for example, a requirement to describe restrictions over European Union countries, or EMEA (Europe, Middle East, and Africa). Geographic restrictions might also be on a finer scale (such as State, County, City, political ward, or even Road). As ever, it is better to allow any standard classification system to be used and even references to locally defined classifications (such as a classification defined according to a named LDAP lookup).

Example:

Permissioning of access to real-time stock market information by exchanges often rules that domestic consumers of the data are required to pay an exchange fee before the information becomes available. Foreign consumers, however, may be granted free access to the real-time data.

3.4.2.2 Verification of Geography

Requirement:

The Language must allow for the specification of acceptable methods for determining the geographic location of an Agent based on whatever criterion is deemed suitable for the application. This is in addition to simply specifying geographic constraints.

Note:

The Internet is a geography-agnostic medium. However, in the physical world, legislators often impose permissioning and access rules. The standard must be sufficiently flexible and extensible to allow the specification of geographic constraints according to many different methods. Note also, that geographic constraints are just as likely to be imposed on intermediaries within the value chain as they are on end-consumers. There are additional issues relating to user-mobility (e.g., a UK registered user may be dialling internationally to a UK ISP from China; rules governing access from China might prove unenforceable because it is impossible to determine the location of the user at the time of access).

Example:

- One application may require geography to be proven by the consumer according to attributes contained within digital certificates issued with reference to a specified Public Key Infrastructure.
- Other applications may rely on the geographic location of the consumers' ISP.

3.4.3 Environmental

3.4.3.1 Predication on any Identifiable Attribute of the Environment

Requirement:

It must be possible within the Language to predicate expressions on any environmental elements and attributes that can be uniquely identified within the syntax and definitions of the Language.

Note:

This includes, but is not limited to, sources of Data Dictionary terms, trusted environments, Rights Enforcement Engines, trusted applications, and data delivery media.

Example:

$RULE_1 \Leftarrow$ if (enforcement-engine = X) then *rights-expression*$_1$

$RULE_2 \Leftarrow$ if (enforcement-engine = Y) then *rights-expression*$_2$

3.5 Requirements Pertaining to Trust

3.5.1 Identification of Trusted Entities

3.5.1.1 Explicit Identification of Trusted Entities

Requirement:

It must be possible to name a trusted entity explicitly as part of a right or obligation constraint.

Note:

Naming is subject to enumerated lists of names being available in data dictionaries. Access control over Data Dictionary access is described in requirement 3.2.7.6.

Example:

The rules embedded into browser software regarding whose digital certificates to accept make a good example of an enumerated list of explicit trusted entities. Similar lists of constraints could be used as parameters to expressions of Rights and Obligations.

3.5.1.2 Chains of Trust

Requirement:

It must be possible to specify a trusted entity by reference to a trusted third party according to some established, and arbitrary, path or chain of trust.

Note:

The standard should not assume any particular technology or method for specifying chains of trust. For example, it would be easy to fall into the trap of expressing the above requirement in terms of Public Key Infrastructures. While PKIs are likely to form the bulk of trust implementations, an inclusive standard would allow for trust infrastructures not yet devised.

Example:

A right of access is granted to anyone with a digital certificate issued relative to a Reuters root CA.

3.5.2 Trusted Time Services

3.5.2.1 Insistence on Trusted Time-Services

Requirement:

It must be possible to specify (optionally) that time-based constraints within a rights expression can be satisfied *only if* a trusted time-service is used to validate the current date and time. If the option is expressed and no trusted time-service is available, the default is to assume that time-based constraints are not met and rights are denied.

Note:

See also Requirement 3.5.2.2 which deals with the explicit identification of a specific time service.

Example:

A rights enforcement mechanism is directed to make reference to a named trusted time service when it evaluates rights expressions rather than simply using local clock information (which may not be trusted since it is under user-control).

3.5.2.2 Specification of Trusted Time Services

Requirement:

Within the general framework for identifying trusted entities, it must be possible to indicate what constitutes a trusted time-service.

Note:

See also Requirement 3.5.2.1 which deals with the general requirement for the use of trusted time services.

Example:

- An enumerated list of trusted services.
- A specification of a digital signature required to identify a trusted time service (i.e., the specification may be a protocol for signing a request for authentication of the trusted time service using digital signatures and chains of trust).

3.5.3 Trusted Applications and Environments

3.5.3.1 Nominating Trusted Applications

Requirement:

It must be possible within the Language to nominate applications to be used for Interacting with an Object.

Note:

Stipulations for defining the mechanism with which to determine trust are given in Requirement 3.5.3.2.

Example:

Only the official Adobe Acrobat Reader is allowed to render this PDF file.

3.5.3.2 Specifying the Mechanism for Trusting an Application

Requirement:

The mechanism for determining that an application is trustworthy must be specifiable in the Language.

Note:

The methods for stipulating which applications are trusted are discussed in Requirement 3.5.3.1.

Example:

- The bytes of the application code are the subject of a hashing function whose result is signed by a specified Trusted Third Party.
- The publisher has signed the application code.
- An auditor, whose signature is countersigned by the publisher, has signed the application code.

- A nominated Web service tests the application dynamically.

3.5.3.3 Trusted Environments

Requirement:

It must be possible within the Language to specify a trusted environment within which both trusted and untrusted applications are allowed to Interact with Digital Objects.

Note:

Stipulations for defining the mechanism with which to determine trust are given in Requirement 3.5.3.2. A generalised approach would recognise levels of trust in applications and environments and allow rules describing combinations of those levels.

Example:

An example trusted environment is the Java sandbox. That environment is designed to protect the user from malicious code. One can imagine modifications to the sandbox's security model which limit (via pseudo operating system services) the ability of untrusted code to perform Interactions on Digital Objects which are disallowed by Rights and Obligations expressions.

3.5.4 Certifiable Audit Trails

3.5.4.1 Obligations on Agents for the Construction of Audit Trails

Requirement:

The Language must contain constructs to allow Agents to specify how downstream Agents must construct audit trails for how they (and other Agents) Interact with Digital Objects.

Note:

Audit trail specifications might include attributes describing the formats and content of audit data, security measures around audit trail management, storage, and maintenance, and access provisions for upstream agents.

Example:

- Billing records are one form of audit trail that may be required.
- Change logs for edited Objects may be required as a consequence of allowing edits to take place.

3.5.5 Agent Authentication

3.5.5.1 Strength of Authentication Credentials

Requirement:

The methods by which Agents authenticate must be specifiable as constraints within the Language.

Note:

This is intended to head off the possibility of security holes downstream.

Example:

- A publisher mandates the mechanism a distributor must employ to authenticate downstream Agents (i.e., this is an Obligation incumbent on the distributor as a consequence of being allowed to distribute the content).
- Rights expressions may be predicated upon the mechanism via which an Agent has authenticated.
- Example authentication methods include (but are not limited to):
    - Username and password
    - Challenge/response
    - Digital Certificate (software)
    - Digital Certificate (on smartcard)
    - Biometric (fingerprint, retinal scan, voice pattern recognition, DNA test)
    - Two or three factor authentication, specifying combinations of the above

3.5.6 Data Integrity

3.5.6.1 Obligations on Agents

Requirement:

The Language must contain constructs to allow Publishers to define constraints on downstream Agents in terms of how they must handle Digital Objects such as to preserve their integrity.

Note:

See also privacy rules discussed in Section 3.5.7.

Example:

- A Publisher requires downstream distribution to take place over tamper-resistant channels (e.g., using an SSL Web connection).
- A Publisher requires distributors to provide a validation mechanism so that consumers can check the integrity of the Digital Objects they receive (optionally using a technique mandated by the publisher).

3.5.7 Agent Mandated Privacy

Requirements in this section pertain specifically to the ability to express, within the Language, rules *about* privacy. See also, Section 3.2.10, which deals with privacy issues as they relate to the structure of the rules themselves, and Section 3.5.6, which deals with data integrity.

3.5.7.1 Privacy of Individuals

Requirement:

The Language must include hooks to make reference to privacy specification protocols such as P3P (see [3]) wherever possible.

Note:

See also Requirement 3.5.7.3 which deals with privacy of organisations.

Example:

As part of the response for acknowledging a contract (see Requirement 3.3.10.2), an Agent is allowed to append to the response a P3P specification of their privacy rights. The workflow may be extended so that the Publisher signs the P3P specification to acknowledge and accept the privacy rights thus asserted.

3.5.7.2 Privacy of Communication

Requirement:

It must be possible to specify the obligation that all transmissions of data are to conform to a particular level of privacy protection.

Note:

There are many ways of ensuring privacy, including channel-agnostic mechanisms such as encryption, the use of private networks, and the employment of strict access control schemes in closed domains. These (and others) may all be used within privacy constraints.

Example:

For Web-distributed data, the originator may specify that all downstream distributors should employ the SSL protocol.

3.5.7.3 Privacy of Organisations

Requirement:

In addition to describing the privacy of individuals, the Language should also allow organisations to make privacy specifications.

Note:

In many legislative domains, individual consumers are far more strongly protected than are commercial (or other) organisations with respect to privacy of their information. However, there is no reason why an organisation may not wish to similarly specify privacy constraints. The constraints may differ from those of individuals because they are likely to refer to concepts such as competitors and market segments. The concept of an organisation is complex.

Example:

- For whatever reason, an organisation may require that it is not identifiable as a customer of a publisher.
- Organisational sub-units may wish to specify privacy constraints with respect to other sub-units within the same parent organisation.

3.5.8 Confidentiality

3.5.8.1 Confidentiality of Source

Requirement:

The Language must make it possible to specify confidentiality clauses that require downstream contributors to respect the confidentiality of the source of the content.

Note:

The obligation might be combined with the general mechanism for contract establishment (see requirement 3.3.10.2).

Example:

A journalist may require anonymity to protect his or her status within an oppressive regime or within a commercial context.

3.5.8.2 Confidentiality of Digital Objects

Requirement:

It must be possible to highlight confidentiality requirements in addition to the usual specification of rights of access.

Note:

Confidentiality covers

- the content itself
- information about who has access to content (i.e., the Rights and Obligations granted to and about content)

Example:

A publisher may have two clients, both of which are granted the same rights of access over a document. However, confidentiality requires that neither can disclose information about the document to the other. This prevents unintentional disclosure of the publisher’s business interests.

3.6 Additional Functional Requirements

3.6.1 Specialised Support for Business to Business (B2B) Commerce

3.6.1.1 Attribute Inheritance

Requirement:

Businesses are inherently hierarchical entities. The Language must allow for Rights and Obligations expressions to be written and evaluated with respect to attributes associated with arbitrary points within the hierarchical structure. Rules may need to be specified describing valid inheritance schemes.

Note:

See also Requirement 3.5.7.3 which discusses privacy constraints for within complex (e.g., hierarchical) organisations.

Example:

- Rights and Obligations granted to all employees of a company (all employees inherit the company's rights).
- Rights and Obligations granted to all employees of a company's treasury function (qualifying employees inherit the departmental rights).
- Rights and Obligations granted to all developers within a company's Research and Development department (the employees role determines whether or not they inherit company-wide, role-based rights).
- Rights and Obligations granted to all developers within a particular team within the Research and Development department (role-based inheritance of departmental rights).
- Rights and Obligations granted to all employees of a certain grade or above (attributed-based inheritance of company-wide rights).
- Rights and Obligations granted to managers with divisional, departmental, or project responsibility (inheritance of rights from various parts of the company hierarchy).

3.6.1.2 Complex Business Models

Requirement:

If the Language is to include business model definitions as part of the description of Obligations then it must accommodate (in whichever manner is deemed necessary) such complexities as:

- Sponsored accounting (where charging is rolled up to specified cost centres),
- Volume discounting based on global accounting, but applied locally,
- B2B-style payment methods such as invoicing.

Note:

It is an open question whether or not such considerations should be part of the Language. However, if charging models are included, then it is a hard requirement to ensure that the full complexity of B2B commerce is covered.

Example:

A document is supplied to an organisation. An Obligation on making access to the document is that payment should be made prior to access being granted. The workflow mechanism that causes payment to be made must interact with the rights enforcement mechanism (if such a thing exists) in order to allow access to complete. Together, the systems must understand that payment by a cost-centre manager on behalf of the individual is sufficient to grant that individual access to the content. If volume discounting is being applied in real-time, then the system must further check and update relevant meters to determine and register the appropriate charge.

3.6.2 Machine Processing of Digital Objects

Three basic models of machine processing and data consumption are relevant:

1. Machine consumption of data for generating derived (value-added) information. Examples in the financial data domain include simple financial ratios at one extreme, or machine intensive technical analysis of trends and fundamentals at the other extreme.
2. Machine processing to allow re-distribution to a different audience further down the industry value chain (value-added distribution). Note that, typically, this model also involves value-added information.
3. Machine consumption of data in order that decisions can be made and actions can be taken automatically. An example is an automatic trading system that attempts to maintain a position by buying and selling shares when preset criteria are met.

3.6.2.1 Data consumption by humans must not be given special status over machine processing

Requirement:

No aspect of the standard must give special status to human consumers at the expense of machine processing.

Note:

The standard should be sufficiently generic to apply to rules governing machine processing as well as to rules governing delivery of data to human “eyes and ears”.

Example:

The inclusion of a rule about rendering devices, which assumes implementation by trusted browser plug-ins, would be an example where this requirement is not met.

4 Other areas for Consideration in Building the Standard

This section is included to capture ideas generated during the construction and research of the Reuters requirements list that are no in themselves requirements, but which might introduce concepts that are relevant to or which influence the standard in some way.

4.1 Channel Definition

Many services, such as news delivery, provide a mechanism by which recipients can tailor the service to fit their needs. In the news example, channels might be defined by a news customer (which may be an end-consumer of news, a distributor, or some other participant in a complex value-chain) according to:

- Topics of interest (e.g., sport, politics, current affairs, financial, regulatory),
- Geographic region (e.g., USA, London, Africa, European Union),
- Importance ranking (according to some pre-defined criterion such as "significant headlines only"),
- and so on, ad infinitum.

The selection of content via channel definition might be viewed as a persistent query against the data source. The delivery of content is shaped both by what the customer is allowed to see and what they have requested. Therefore, there is interplay between permissioning and channel definition systems. It is most likely that channel definition will remain outside the scope of rights markup. However, given the functional interplay between the two, it is desired that the rights markup is mutually compatible with whatever channel definition languages are available. The wider MPEG-21 standard may choose to employ or define a recommended channel definition language for this purpose.

A variation on channel definition is for a publisher to specify channels according to attributes of a recipient. For example, different natural language variants of a document might be made available according to geography or registration details of a customer. This may be related to permissioning information and may therefore have relevance for the rights Language.

4.2 Object Models

In defining Rights and Obligations over value-chain participants, content consumers, and so on, it may be useful to build an extensible object model to classify and formalise relationships between the participating entities. An example where object models have proven useful in formalising rights descriptions is the ODRL specification [4].

An extensible object model could form an integral part of the Rights Expression Language. The model may distinguish, for example, between human and machine consumers of content, allowing Rights and Obligations expressions to be written in terms of the distinct classifications. The object model could be used to define what is meant by a role, or how entities may act on behalf of others as proxies.

Organisational structure could be represented within the object model. The model might describe how qualifications could be inherited within an organisational structure. For example, where a right is conveyed upon an organisation, there may be rules which define how the right maps onto different classes of membership within the organisation. The object model would provide a basic framework onto which the organisational structure can be mapped and against which criteria can be specified to define the inheritance model for qualifying attributes.

In this section, the discussion has centred on an object model of participating entities. This is by no means the limit of application for object models (the ODRL specification, for example, applies object modelling to many other aspects of a rights system). It is worth noting that, without such a formalisation, it may prove difficult to reason about many of the more complex ideas raised by the requirements of this document. Finally, any object model should be extensible and would, ideally, be represented as part of the Rights and Obligations Data Dictionary.

4.3 Workflow

There are many aspects of workflow automation that are relevant to rights and obligations management. Workflow may be entirely contained within a single organisation or it might be extended to encompass tasks undertaken between numerous, collaborating organisations. In both cases, rules are needed to describe what is allowed and the sequences that should be followed.

One observation recently described to the author is that planned workflow can be an alternative solution to access permissioning. The argument proceeds as follows. Where access to Digital Objects is controlled by rules governing Rights and Obligations, both individual Interactions and sequences of Interactions between Agents and Objects are controlled. A defined workflow represents Interaction sequences in an alternative manner, stating precisely the order of Interactions. A single rule: "follow the workflow definition" effectively limits Interactions in the same way as might be achieved with separate Rights and Obligations expressions. Whether or not this is a valid argument, it is clear that there is a close relationship between workflow and rules for Rights and Obligations.

Consider the case of inter-organisational workflow, designed to support automation of B2B exchanges. Figure 5 shows three models for achieving integrated workflow between organisations:

(i) via Standard Interfaces (e.g., using a workflow markup language)

(ii) via a workflow mediation service (which understands the workflows of both organisations and translates between them)

(iii) by stretching the boundaries of one organisation's workflow to reach inside another organisation.

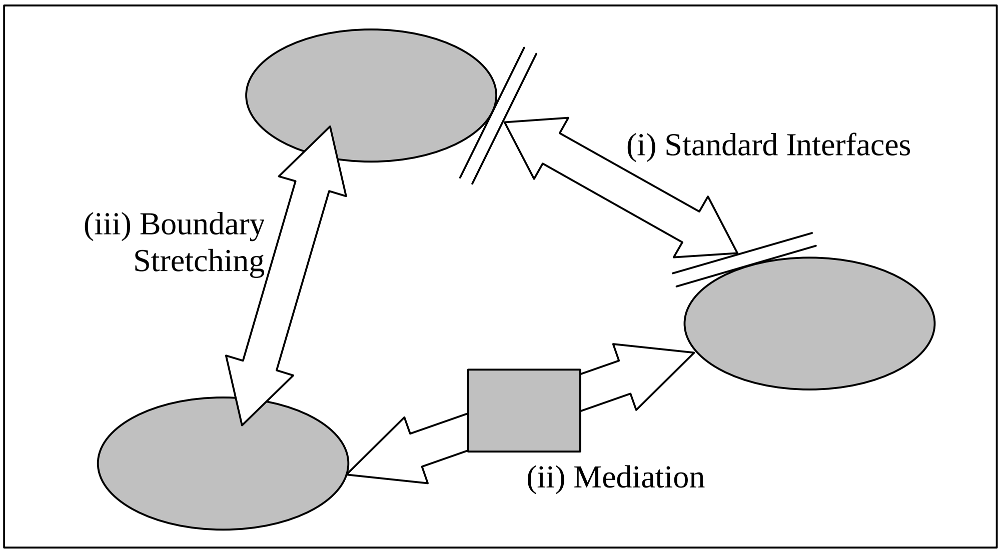


Figure 5 Three Models for Integrating B2B Workflow

Whichever of the above mechanisms are used, once can see the need for defining Rights and Obligations at the interface points. Certain of the requirements in this response have alluded to workflow (e.g., in the sequencing of Obligations). Where workflow languages exists, due consideration should be given to the part those language have to play in defining a Rights Expression Language. The Workflow Management Coalition (see http://www.wfmc.org/) is actively promoting workflow standards and has created a Workflow Reference Model [10].

5 Summary and Conclusions

In summary, this response has listed a large and varied collection of requirements pertaining to the expression of digital rights and obligations. It is not an exhaustive set of requirements, but every attempt has been made to cover as much ground as possible. It is acknowledged that there may be some redundancy in the requirements listed.

It is clear, simply from attempting to build a list of requirements around the MPEG-21 call, that the problems being addressed by the formation of a standard Rights Data Dictionary and Rights Expression Language are not insignificant. The task of building a language that is sufficiently flexible and extensible to avoid having to rewrite the standard in the very near future should not be underestimated.

If MPEG-21 is to address the needs of the global community it has an obligation to build the standards as wide and encompassing as possible. A policy of avoiding exclusions (i.e., one of seeking the most general case) is encouraged. Clearly, the standards are of little practical use if they cannot form the basis of implementations. However, it has been recognised that merely being able to express contractual terms and conditions in machine-readable form is a large step forwards in being able to manage effectively intellectual property rights and obligations in the new digital world. The ability to create a unified model of permissions, entitlements, obligations, and underlying legal contracts is one that will bring benefit in terms of building complex products, utilising the resources of many suppliers.

Implementations of schemes that enforce and take account of rights and obligations expressions will be based around the three complimentary approaches of technical, business, and legal solutions. Just as e-commerce cannot function properly without an e-business strategy; intellectual property rights management in the digital domain requires a strong business strategy, supported by technical implementations and a consistent legal framework. Also, there is much to be said for simply explaining, in clear, precise terms, the rights and obligations placed upon business partners and consumers. Technical enforcement may not always be required. Audit processes, contract law, and common good business practice provides an excellent framework in which to conduct business digitally. These should not be forgotten in the rush to protect intellectual property. Moreover, schemes for detecting misuse of IP have a fairly significant part to play in the digital rights management arena.

The task ahead is monumental. However, if the goals of MPEG-21 are achieved then there is potential to unlock the vast resources that have never been made available for digital consumption for fear that the IP would instantly become worthless.

Appendix A Acronyms and Glossary

In addition to the terms and acronyms defined in the following table, the reader is also referred to the definitions given in Section 1.6 of the main response document.

ACL Access Control List—a mechanism for limiting access to content. An ACL is bound tightly to a resource and lists explicitly the entities granted and/or denied access to the resource.

Agent Term defined in Section 1.6.1.6.

B2B Business-to-business.

B2C Business-to-consumer.

B2E Business-to-employee.

B2G Business-to-government.

CA Certificate Authority (a Public Key Infrastructure term).

Contact Term defined in Section 1.6.1.5.

Context Term defined in Section 1.6.1.7.

Digital Object Term defined in Section 1.6.1.2.

DRM Digital Rights Management—The definition, protection, and/or enforcement of rights pertaining to Content produced, delivered or accessed electronically.

HTTP Hypertext transfer protocol—the mechanism by which Web browsers communicate with Web servers in order to retrieve documents and follow hyperlinks.

HTTPS A secure version of HTTP employing SSL technology.

Interaction Term defined in Section 1.6.1.5.

IPR Intellectual Property Rights

ISP Internet Service Provider—a utility business providing access to the Internet for consumers and businesses.

Language Term defined in Section 1.6.1.1.

LDAP Lightweight Directory Access Protocol—a cut down version of X.500

P3P Platform for Privacy Preferences—a W3C initiative.

PDF Adobe’s Portable Document Format.

PKI Public Key Infrastructure.

Publisher Term defined in Section 1.6.1.9.

RDD Rights Data Dictionary.

RDD-REL Rights Data Dictionary and Rights Expression Language.

REL Rights Expression Language.

Rights Data Dictionary Term defined in Section 1.6.1.4.

Rights Language Term defined in Section 1.6.1.4.

SSL^1 Source Sink Library—a Reuters acronym coined prior to the more commonplace Internet acronym (see SSL^2). References to SSL in this document refer to the non-Reuters version of the acronym.

SSL^2 Secure Sockets Layer—the more common version of the SSL acronym describing a communications protocol which uses encryption to protect data content over a peer to peer connection. The HTTPS Web protocol makes use of SSL between Web servers and browsers to implement a secure version of HTTP.

Standard Prelude Term defined in Section 1.6.1.3.

Token Term defined in Section 1.6.1.8.

TTP Trusted Third Party—in the context of PKI, an entity implicitly trusted by organisations and end users, such as a government agency. The TTP countersigns, and thereby legitimises, certificates issued by Certificate Authorities. TTP is therefore a root CA.

URI Uniform Resource Identifier—an extended form of URL.

URL Uniform Resource Locator—the primary mechanism for addressing content/resources on the World Wide Web. Examples are http://www.reuters.com and `ldap://x/y/z`.

VPN Virtual Private Network—a “network within a network” established using *tunneling* protocols. Typically a public network such as the Internet is used to carry traffic that is protected by a layer of encryption. Only legitimate participants in the virtual network are able to make sense of the encrypted traffic.

W3C World Wide Web Consortium—a standards body dealing with Web-related Internet standards, such as HTML, XML, and XML-derivative languages.

WfMC Workflow Management Coalition (see http://www.wfmc.org/).

X.500 A standard for directories.

X.509 A standard for digital certificates.

XKMS XML Key Management Specification (see http://www.w3.org/TR/xkms/). XKMS defines protocols for distributing and registering public keys, suitable for use in conjunction with the proposed standard for XML Signature [XML-SIG] developed by the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) and an anticipated companion standard for XML encryption.

XML eXtensible Markup Language.

Appendix B Bibliography and Web References

[1] *Call for Requirements for a Rights Data Dictionary and a Rights Expression Language (Re-issue), March 2001. MPEG-21*. See: *http://www.cselt.it/mpeg/cfp/call_for_requirements(rights_language).htm* on the Web.

[2] NewsML Functional Specification Version 1.0. International Press and Telecommunications Council, 24 October 2000.

[3] Platform for Privacy Preferences (P3P) Project. See: *http://www.w3.org/P3P/* on the Web.

[4] Open Digital Rights Language (ODRL) v0.8, 21/11/2000, ed. Renato Iannella, IPRSystems Inc., (mailto: *renato@iprsystems.com*). See *http://odrl.net/ODRL-08.pdf* on the Web.

[5] W3C Web Services. See *http://www.w3.org/2001/01/WSWS* for an overview on the Web.

[6] The Publius (peer-to-peer publication system) Web site is at: *http://www.cs.nyu.edu/waldman/publius* on the Internet.

[7] Namespaces in XML. W3C Recommendation, January 1999. See *http://www.w3.org/TR/1999/REC-xml-names-19990114/* on the Web.

[8] XML Key Management Specification (XKMS) Submission Request to the W3C. 30 March 2001. Publicly available on the Web at: *http://www.w3.org/Submission/2001/08/*.

[9] eXtensible rights Markup Language (XrML). ContentGuard Inc. See *http://www.xrml.org* on the Web.

[10] The Workflow Reference Model of the Workflow Management Coalition. See *http://www.wfmc.org/standards/docs/tc003v11.pdf* on the Web.

[11] ebXML Web site; see *http://www.ebxml.org/*.

[END]