Metadata models for electronic reading

Olli Alm, Siqi Hao, Jörgen Eriksson
Helsinki Metropolia University of Applied Sciences

Version history

Version	Date	State	Editor	Notes
1.0	3.9.2010	Initial Version	Olli Alm	–

Contributors

Name	Contact	Organization
Olli Alm	Olli.Alm@Metropolia.fi	Metropolia
Siqi Hao	Siqi.Hao@Metropolia.fi	Metropolia
Jörgen Eriksson	Jorgen.Eriksson@Metropolia.fi	Metropolia

Contents

1 Introduction

This report investigates the pertinent metadata models related to the field of electronic publishing and delivery, namely *eReading*. In the report, we first describe the concept of eReading and the essential parts and building blocks of the electronic publishing.

The project's scope is to study the book, magazine and newspaper domains of eReading. The perspective is technical: what functionalities are needed in order to publish, retrieve, share and consume the readable material for electronic reading devices. After describing the overall process, we define the concept of metadata model and present some of the most prominent domain models.

2 What is eReading?

As long as there has been computers, there has been textual content consumed from the computer screens. From the mid 1970s, there has been standards for representing graphical page images in the electronic format[1]. Adobe's Portable Document Format (PDF), the current standard document format for facsimile copies of printed material has been available from 1993[2]. In addition to document formats with fixed layout, another de facto standard for document representation has been HTML in WWW. The amount of electronic material in the web pages is huge. In 2008, Google announced that they have 1 trillion unique URLs in their search index[3].

Within last few years, there has been new rise of eReading, in the form of eReader devices. The latest era of ereaders, started from the success of the Amazon's Kindle in 2007[4], has provided a boost of existing reading infrastructures by better usability, mobility and easy access via wireless networks. If the concept of eReading is defined by the approaches or development offered by the latest eReader devices, we found at least following definitive features for the concept:

- **Special devices**: Emphasis to provide the content for lightweight and mobile reading devices to provide good usability and user experience. (e.g. eInk devices like Amazon Kindle, B&N Nook and Sony reader. Tablet PC's like Apple's IPad and Tablets based on Android OS.)
- **Representation**: Content provided for the eReaders imitate the experience of printed content. Especially, content is splitted into pages instead of having (scrollable) over-screen container area.
- **Quality screens**: ereader manufacturers provide quality screens with high contrast and paper-like feeling (e.g. eInk technologies).
- **Business**: Emphasis to develop platforms for buying, consuming and managing the content online and on-demand. Crucial aspect for the success of eReading is the well founded, profitable business models.

[1] Postscript format, http://en.wikipedia.org/wiki/PostScript
[2] PDF, http://en.wikipedia.org/wiki/Portable_Document_Format
[3] http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html
[4] see http://en.wikipedia.org/wiki/Amazon_Kindle

If we make an assumption that the essence of eReading something different than WWW, it should be clear that it is not defined by the amount of material. Also, there has been books, articles and newspapers available for a long time in electronic format. It seems that eReading is not about representing the material in the electronic format, it is more about providing and imitating the infrastructure of the printed media in the electronic format, including payment issues, accessibility, reaching the material, copyright / legislative issues and publishing. The idea is to investigate opportunities provided by the new technological solutions, and enhance at least the presence and availability of the content.

2.1 Publishing process for eReading

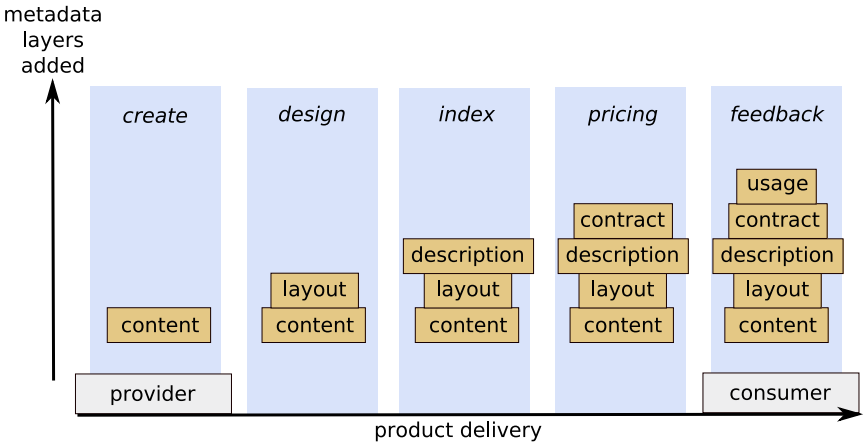

Figure 1: Publishing process and related metadata for eReading

The basic scenario of the eReading publishing process is illustrated in figure 1. Starting from the left side, the content production starts from getting the content or converting it to the proper electronic format (*content*). The second phase is to do the design for the content *layout*. The third phase is to describe the content (*description*). This means that the descriptive metadata fields of the schema are filled in order to be able to search and retrieve the material from the indexing service. By the indexing, we mean the technical operation to store the material for the retrieval functionalities, including keyword search, recommendations, profile matching, categorical or faceted search. To provide the content for the consumer, we have to define the price and the terms of use for the content (*contract*). Depending on the system architecture and the DRM approach followed, the system may force the customer to follow rights restrictions. In the chapter 5.3, we will introduce the ODRL approach for this. At the customer end (or at the client side), the content statistics will be collected and the feedback about user behavior may be returned to the content provider (*usage*).

2.2 Types of metadata related to eReading

By the term *eReading*, we refer to all kind of readable content in electronic format. In the project scope, the eReading is splitted into three different domain

areas: 1) Books, 2) Magazines and 3) Newspapers. For all areas we have similar metadata features. Based on the NISO's article [2] we present the following generic classification for different types of metadata:

- **Content and representation**
  - **Content** text, pictures, "atomic" units of the content
  - **Structure** building the flow of the content from the atomic content units
  - **Layout** how the content is represented and shown in the user interface
- **Adminstrative metadata**
  - **Authorization** DRM, pricing, copyright, regional information
  - **Preservation** metadata for long-time storing and archiving of the content.
- **Content description**
  - **Keywords**
  - **Publication metadata** date issued, version, original publishing date, language
- **Technical information** file formats, MIME-types, protocols used
- **Usage** statistics, user tracking

Universal Identifiers
In addition, all the objects modeled are having unique identifiers. On many cases, these is a need for *universal identifiers* that remain stable (do not vary at all) and can be utilized as a reference points from multiple systems. For universal identifiers, there all many *identifier schemes* that define their purpose, usage and construction.

- **URI, URN, URL** Universal Resource Identifier / Name / Locator. Identity framework for WWW by W3C[5]
- **ISBN, ISSN**
- **DOI** Digital Object Identifiers, upcoming ISO-standard digital material identification[6].

To provide interoperability of the material for different reading platforms, it is useful to agree on the practices on what identifier schemes should be utilized and how. For example, ePub and NewsML promote URIs as identifiers for publications.

[5] see http://en.wikipedia.org/wiki/Uniform_Resource_Identifier
[6] see http://www.doi.org/faq.html

3 What are metadata models?

In this chapter we give a short introduction to metadata models and their components (*schemas*, *vocabularies*) and building blocks: *objects*(classes), *properties* and *value ranges*. We define metadata models as informally or formally defined models to describe information of a specific domain of interest. A metadata model defines the set of things that should be useful for describing the things to be modelled. Metadata model is a *meta* model, above the actual information: it defines the inventory of elements that may exist in the domain.

3.1 Metadata of metadata models

The basic unit of metadata models is *schema*. Schema defines the kind of *objects* (classes) (1)that exist in the domain. In addition, schema defines *properties* (features) (2) of the objects and *value ranges* (3) for properties. To put everything together, the schema also describes the connections between the objects and properties, and between properties and value ranges.

In the following, an example model concerning the book domain is presented. The model consists of following elements:

1. **objects:** book, person, subject
2. **properties:** name, author, no of pages, label, language, nationality, keyword
3. **value ranges:**
   - **atomic datatypes** positive integers, strings
   - **predefined, controlled vocabularies** languages of the world (e.g. ISO 639-1[7]), list of nationalities (e.g. ISO 3166-1[8]. ISO 3166-1 is a listing of countries: nationalities without countries may not be included in the list.)
   - **objects of the models** e.g. book's author is *a person*

The connections between the elements are depicted in the figure 2. Each square represents an object. Inside an object, the properties are listed. For each property, the value range is defined. The value range may refer to (other) objects (*person*, *subject*), specific (atomic) datatypes (*string*, *integer*) and predefined vocabularies.

By *vocabulary*, we refer to a predefined, controlled collection of terms or concepts. By the Semantic Web movement[1] there has been new rise of common, reusable vocabularies. In the Semantic Web context, the vocabularies have been modelled as domain ontologies, where the terms of vocabularies form an interconnected network of concepts. In this report we use the term *vocabulary* as a generic notion to refer the approaches where set of terms or concepts are utilized as a inventory of objects of a specific area of interest, covering catalogs, terminologies, (controlled) vocabularies, glossaries, thesauri and (domain) ontologies[9] Different kind of vocabularies are suitable for describing the content, but more formalized and interconnected vocabularies may provide better

[7] http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
[8] http://en.wikipedia.org/wiki/ISO_3166-1
[9] For further information about classification of the vocabulary approaches, see [20, 11].

support for automated reasoning or other tasks related to the field of symbolic artificial intelligence. The common factor for all vocabulary approaches is that 1) items in the vocabulary have the same type (e.g. term, concept) and 2) the items usually cover widely the subject area in matter. The editorial work and publishing may be controlled by an author, usually the professionals of the subject (*controlled vocabularies*) or be more free (*tag vocabularies*) and user generated (*folksonomies*).

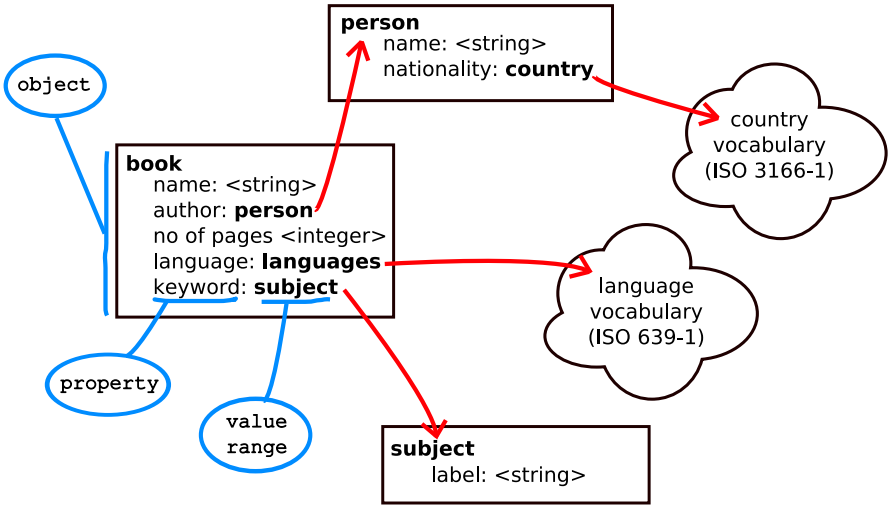

Figure 2: Simple metadata model for books

A metadata model may be informative or descriptive; it doesn't have to have actualization on a specific formal language. Instead, we may have a metadata model that has implementations on different formal machine-readable languages. A formal implementation of a metadata model may adjust or specify the unformal description or follow the description in different ways. In addition, the expressions in the formal language may be interpreted in multiple ways by the machine.

3.2 Metadata for interoperability

The metadata models provide a unified approach for modeling some domain of discourse. Even without committing to a specific model, a model may provide good guidelines for designing a system or a data model. If a specific implementation (e.g. in XML or RDF formats) is utilized, it provides a way to interoperability with the system. The interoperability may be implemented on a several levels. For example, by utilizing a specific (standard packaging and delivery) file format we have an approach where set of applications may utilize the same kind of objects heedless of the way the files are delivered between the applications. Standard file formats for books such as EPUB and PDF support this approach.

The provision of commensurable content can be provided by

1. utilizing same or connected metadata schemas

2. utilizing same or connected domain vocabularies

3. or having both connected vocabularies and schemas.

For example, to provide an interface for searching both newspaper articles and books, we should define how the newspaper articles relate to individual books. On the schema level, the mapping can be done by using *same* properties (title, author, date issued) for both objects or by mapping together the corresponding object properties (book title - headline, author - reporter, publishing date - date issued). By connected schemas, we can retrieve different kind of objects (news articles, books) corresponding to a specific property values defined by the query. The schema-mapping problems arises when we have a set of diverse objects (persons, audio, video, books) and a variety of properties that exist only with part of the objects (street address, phone number) or having unconnected value ranges (length of the book vs. length of the video).

By utilizing common vocabularies or value ranges we can provide retrieval scenarios where we get every object that are related to vocabulary but do not know *how*. For example 'date and time' value range may be connected to different objects with properties with diverse ways: date of publication, last accessed, last modified, revision date, etc. The value range modeling might also involve complex values: 'date and time' may refer to a specific point in (linear, calendar) time or specific time span (duration of one week). In addition, complexity grows with cardinality issues: an object may allow only one value or multiple values. In the following list some example vocabularies are provided:

- ISO 639-2, language name vocabulary. Lists languages and their corresponding codes. Example terms: 'fin' for Finnish.
- NewsCode descriptive vocabularies provided by IPTC:
  - Genre. Defines the type of a news item. Terms: analysis, forecast, review, quote.
  - Media Topic. Defines the topic of an item. Terms: fashion, road incident, energy industry, sport.
  - Scene. Defines the category of what is represented in a picture. Terms: aerial view, night scene, couple (two persons).
  - Subject code. Description of the content on three levels of abstraction. More detailed than media topic, ca 1400 terms. Terms: trials, sculpture, livestock farming.
  - Subject Qualifier. Adjusts the subject code definition. For subject code sport, we could have qualifiers woman and outdoor.
  - World region: List of continents. E.g. Europe, Asia.
- Magazine vocabulary on Onki Ontology server, developed in the Crossmedia project. [9, p.26]
- Kaunokki vocabulary for fiction books, in Finnish
- Prism controlled vocabularies in RDF-format. For Prism, see chapter 6.1.

4 Models for eReading content

In this section, we go through of some of the promising or widely used models for eReading content. These models describe how a content for a specific purpose should represented and described. Currently, the main source for these models are the book domain (ePub, PDF) and news domain (NewsML, RSS and ATOM).

In the book domain, there exist variety of format for eReading devices. Some main formats, such as Mobipocket, ePub, Amazon's AZW are derived from the Open eBook standard and they use the XHTML as a standard representation format.

On the academic field, especially in the field where the mathematical notation is needed, latex and bibtex formats are widely used. Latex is a content language for defining the layout and structure for a text document.

4.1 EPub

Epub is a standard format for ebook representation, delivery and packaging. It is provided by the International Digital Publishing Forum (IDPF)[10]. The basic scenario of the ePub is to provide same content for different kind of devices by reflowable content layout. Because the ePub packaging and file format, it is suitable for offline usage and custom reader applications. Epub consists of three separate definitions:

- **OPS** defines what and how file formats are utilized inside ePub package.
- **OPF** defines the content, its structure and metadata.
- **OCF** defines how to bundle the content and its definitions to a transferable object (a zipped file).

4.1.1 Open Publication Structure (OPS)

OPS specification[19] defines the file formats utilized in the ePub packages and how the applications should process and interpret the formats. The scope of the specification is rather technical, including definitions for file encoding formats, XML well-formedness, supported MIME-types and processing and rendering (how the content should be displayed) instructions. OPS defines also the subset of HTML elements allowed for content representation.

A noticable feature of OPS specification are the recommendations for accessibility. Because ePub is utilized in variety of devices, the OPS allows additional semantics but do not require them to be interpreted. For example, the content provider may provide additional formatting information inside ePub file for special devices suitable for visually impaired. OPS specification also recommends authoring the the HTML content according to W3C's web and mobile device accessibility guidelines.

[10] http://www.idpf.org/

4.1.2 Open Packaging Format (OPF)

OPF defines the individual content files (e.g. book chapters), their mutual relations and the publication metadata. The main components of OPF document are the following[18]:

- **metadata** description of the content.
- **manifest** list of files.
- **spine** linear reading order of the manifest files
- **guide (optional)** reference for essential structural subcomponents of the book, e.g. cover, table of contents, glossary, list of figures (loi). For full listing of types, see [18], chapter 2.6.

The publication **metadata** consists of following properties defined by Dublin Core Metadata Element Set[4]:

- **title** name of the book ("Moby Dick")
- **creator** main author ("Herman Melville")
- **subject** keywords describing the content ("sailor", "whale")
- **description** free-form description of the title ("the adventures of the wandering sailor Ishmael, and his voyage on the whaleship Pequod")
- **publisher** a person or organization making the resource available ("Richard Bentley")
- **contributor** additional contributors: editor, designer, illustrator
- **date** date of publication. With an additional attribute, the date can be further defined to be date of modification or creation. ("1851-10-18")
- **type** category or genre of the title ("novel", "fiction")
- **format** primary media type of the content, usually derived from MIME media types[12, 13]("application/xhtml+xml")
- **identifier** book identifier; the identifier scheme is not fixed: URIs, URNs, DOIs or ISBNs or custom identifiers can be utilized. With an optional attribute 'scheme' the information of the identifier scheme can be provided. ("123"),
- **source** reference to previous or original version of the content
- **language** primary language, derived from RFC3066[10] or its successors (e.g. BCP 47[11]). ("en")
- **relation** related or auxiliary publications, e.g. printed version of the book. ("ISBN-XXX")
- **coverage** defines the scope or coverage of the content, e.g. geographical market area or temporal period for selling the item

[11] http://tools.ietf.org/html/bcp47

- **rights** copyright notice or other informative description of the usage rights; insufficient for defining how the content should be secured

With the special attributes, the field **contributor** can be further specified with the role derived from MARC relator code list[12], e.g. having role of *editor*, *artist* or *illustrator*.

For custom, additional metadata the latest OPS draft[19] proposes the usage of XHTML1.1 *meta*-elements inside *metadata* component:

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:opf="http://www.idpf.org/2007/opf">
   <dc:title>Tale of Two Cities</dc:title>
   <dc:creator opf:role="aut">Charles Dickens</dc:creator>
   ...
   <meta name="price" content="USD 19.99" />
</metadata>
```

In order to exploit custom metadata fields (such as 'price'), the ePub application have to be aware of their meaning and way they should be processed.

The content description model of the ePub format defines the usage of the properties and their value ranges informally; the spefication itself is insufficient for proper, automated machine processing of the content. With the sophisticated metadata editor and guidelines for describing the it is possible to unify the way the books are described. For example, an editor may provide an user interface for 1) forcing the usage of predefined value ranges, 2) validating the syntax and proper usage of the user input, 3) recommendation of suitable values and 4) filling automatically fields based on the content and contextual information.

4.1.3 Open Container Format (OCF)

OCF specification[17] defines how the files describing the ePub publication have to be encapsulated within a one container file. It define the names of the files, encodings, the directory structure included in a ePub package.

The single-file storing approach is ideal for easy transportation, delivery and offline usage. The zip packaging format is used as a container file format for ePub book packages.

4.1.4 Summary of Epub 2.0

Epub format is a standard representation for delivering, packaging and representing books. The layout strategy is based on the reflowable approach and it supports different screen sizes and accessibility by supporting font size increase. The content represented by OPF consists of invidual building blocks (e.g. chapters of the book) that by default have one and only one linear order[13]. The linearity suits well for novels, but for a certain kind of factbooks, there might be need for representation where the structure is not totally fixed in a sequential order.

OPS defines a set of media types that has to be supported by the ePub system. The supported types consists of image formats (gif, jpg, png, svg),

[12] http://www.loc.gov/marc/relators/

[13] In addition, objects can be marked not to be shown in as primary objects, without ordering

XHTML for content representation, digital talking book[14], CSS for HTML formatting and XML [19]. With the OPF, the ePub can provide content in any kind of media files as long as there is fallback item provided. A fallback item provides alternative version of the item with approved Mime-type. For example, content can be presented as a PDF, as long as there is also the HTML-version of the content for those devices or applications that cannot understand PDF-format. In practice this means, that OPF format provides means for representing any kind of content as a linear sequence.

4.1.5 EPub 2.1

Currently IDPF has launched a working group for developing new version of the Epub. In the latest working group charter (May 2010):

- **Extending the domain** instead of supporting books, the idea is also to support magazines and newspapers by supporting articles as atomic units (3). Support for advanced non-linear structures for academic publishing, support for mathematical symbol representation (9, 10).
- **Better user experience** support for rich media components (1), page-specific and dynamic layout (5), enhanced navigation with NCX[15] (6).
- **Support and extension of metadata** Support for existing languages for content description (4), alignment of Epub with existing web standards (7), support for industry specific extensions (12).
- **Accessibility** better support for synchronization of text and audio (11).
- **New functionalities** Support for advertisement (14), support for annotations (8).
- **Localization** Support for global languages and typographies (2), defining the relation to national and international standards (13).

With the new additions, there is a danger that the complexity of the Epub increases. For example, by adding rich media support inside standard proposal, the language may easily approach a fully fledged programming language. If the rich media implementation is based on HTML5, is there still clear need for ePub?

In the working draft it is mentioned that the DRM issues have been left out from the proposal. The decision is reasonable: from the perspective of open, file-based delivery format it is difficult to overcome the DRM security and contracting issues.

Initial working draft should be available in September 2010, final standard recommendation in May 2011[6].

4.2 Portable Document Format (PDF)

Adobe's PDF document format provides simple and neat schema for describing the content. All the values are optional and the value ranges are defined with

[14] http://www.niso.org/workrooms/daisy/Z39-86-2005.html

[15] Hierarchical, tree-like navigation structure standard defined in Daisy standard[22]

Property	Datatype	Description
Title	string	Title of the document
Author	string	Person who created the document
Subject	string	Subject of the document
Keywords	string	Associated keywords
Creator	string	Application that created the document
Producer	string	Application that converted the document to PDF format
CreationDate	date	Date and time of creation
ModDate	date	Most recent date and time of modification
Trapper	name	Trapping information to support correct printing; see [8, p.974]

Table 1: PDF metadata fields for content description[8, p.844]

the basic datatypes (string, date). The schema is presented in table 1[8, p.844]. For richer description of the content, Adobe has provided XMP platform to embed metadata information in different (binary) file formats. Currently XMP is utilized mainly in Adobe's own products[16]. The XMP specification [7] provides examples for embedding metadata with the XMP-mechanism into different picture-formats (png, svg, gif, jpeg) and document formats (HTML, PS, PDF).

4.3 RSS and ATOM

RSS (Really Simple Syndication) and Atom are languages for delivering and publishing news in the WWW as web feeds. Web feed is a timely oriented publication channel for providing news. Web feeds can be read with client applications that list the latest set of news for the end user. Both feed formats are defined in XML-format.

RSS 2.0
The latest RSS specification is 2.0. The main components of the RSS are the *channel* and *item*. Channel defines the purpose and identity of a specific feed, items are the individual news objects provided by the channel. The basic metadata elements of a channel describe the purpose and management information for reading a channel[5]:

- **Descriptive metadata** title, description, language, category
- **Adminstrative metadata** copyright, managingEditor, publication date
- **Management metadata** latest update (lastBuildDate), unavailability of the channel (skipHours, skipDays)

For an item, the description contains the actual content of the news or synopsis. For reaching the whole news story on the original web page, the link is provided in the item. With the *enclosure* element, the items may contain any kind of

[16] See http://www.adobe.com/products/xmp/, http://www.adobe.com/products/xmp/related.html, http://www.adobe.com/products/xmp/standards.html.

media objects. With the enclosures, the content provider may provide simple multimedia feeds: the feed cannot define how the enclosure elements should be shown or structured with the text.

The most essential descriptive elements for an item are *category* (e.g. tags or terms from controlled vocabulary) and *publication date*. With the metadata the client application can filter out specific set of news related to keywords on a specific time span. Because a client application may consume multiple feeds at the same time[17], The RSS feed mechanism provides a powerful mechanism for tracking the latest things relating to a specific subject matter.

Atom

The Atom syndication language[23] provides the similar main functionalities as RSS2.0. In Atom, channels are called *feeds* and items are called *entries*. The main reason for developing Atom format was to overcome the vagueness of the RSS[18]. Different RSS implementations has caused interoperability problems. In RSS, there is no way to indicate the format used for textual content description.

For the content description, Atom provides a way to separate the synopsis from the full news story (*summary and* content elements). Atom also supports multilingual feeds by defining the language for indivual items. With the item-specific language definition, the user is able to read content with her own language only.

Both of the language are very similar. RSS is more widely adopted. Atom provides some extra features and unambiguous, standard way for describing elements. The syndication languages provide scarce support for content layout: for this reason a client application should provide layout templates for content representation. In the eReading content, RSS and Atom are languages for delivering and customizing news content for the reader platforms. For the offline usage, the feed should be able to provide the whole content and links for the external sites should be disabled.

4.4 NewsML

NewsML is a news exchange and transportation format for delivering news. As a delivery format it can be applied for B-to-B connections for editorial systems, news agencies, publishers and news aggregators. It also suitable for delivering the format from the service providers to end users. NewsML standard and its successor NewsML-G2 are developed by IPTC (International Press Telecommunications Council) and are also their registered trademarks[19].

The NewsML XML-file consists of following main components[21]:

- **NewsEnvelope** transmission information. Who sent the XML-file itself for whom, and when.
- **Identification** release information of the XML-file by the content provider.
- **NewsManagement** what kind of content is within this delivery and is it ready to be published.

[17] In addition, a feed itself may consume other feeds and provide a feed aggregated from diverse sources.

[18] see http://www.intertwingly.net/wiki/pie/Rss20AndAtom10Compared

[19] http://www.iptc.org/cms/site/index.html?channel=CH0087

- **NewsComponent** for each individual newsitem inside the XML-file, the following information can be provided:
  - **NewsLines** Short, human-readable description of the content. May contain the headline of the newsitem, keywords and date.
  - **Admistrative Metadata** Information about the content provider.
  - **Descriptive Metadata** Description of the content. E.g. language, keywords, target audience.
  - **Content Item** Actual content and its format information. May contain text, audio, video, picture, etc. The textual content is usually presented in XHTML or NITF[20] format.

An individual XML-file defines the prime unit for news management, *a news item*. It is the unit of interchange in a news publishing and delivery cycle. A news item might represent 1) a unitary resource (text, photo, video clip), 2) a multimedia package (e.g. text, photos and thumbnails of an article) and 3) a collection of related news items (e.g. a set of related articles, including text and photos)[21].

NewsML is well specified language with broad range of features. For authoring the news content, NewsML provides status codes for postponing or cancelling the publication of delivered news. Moreover, the status can be triggered automatically on a specific date, e.g. to be usable after a specific date [21]. NewsML offers rich definitions for interlinking separate newsitems. News can be defined as being *derived* or *updated* from the previous items, or they can be associated as a series of articles.

Along with the actual NewsML language, IPTC provides a set of controlled vocabulary for describing and authoring news content. The specification[21] defines best practices for utilizing and extending the vocabularies.

4.4.1 NewsMLG2 and other G2 languages

On 2008, IPTC provided the new version of NewsML along with two special sublanguages: EventsML and SportsML[14]. G2-languages is an extensible and flexible family of languages for describing news content. The language is content neutral, not restring the content or file formats utilized[14].

EventsML and SportsML provides a binding of the model to the ideas of the Semantic Web[1], they are models for describing real world objects that can be referred from the news. From the information retrieval perspective, the events act as aggregators for the news, providing an alternative, spatiotemporal retrieval of the content. Even though that the idea of events layer is nice, it may provide an additional and unnecessary metalayer between the news item and objects they are describing. The actual news may have redundant or duplicate information with the events. The previous version of NewsML provided a linkage mechanism to connect individual news without additional event-metalayer. The usage of additional descriptive metalayers should be carefully weighted: what are the benefits for utilizing complex metamodels.

SportsML is the another sublanguage of the NewsML. We can say that is describes also real world objects, such as teams, players, results, schedules and

[20] News Industry Text Format, an XML standard for defining the structure and metadata for independent news articles. See http://www.iptc.org/cms/site/index.html?channel=CH0153

venues. However, these objects and statistics related to them are relevant news content as themselves, they are not constituting the additional metalayer for the sports related news.

5 DRM

5.1 DRM and DRM systems

Digital Rights Management (DRM) refers to technological means to impose the way digital material is utilized. In practice, DRM technologies provide solutions to restrict the (re)delivery or spreading the copyrighted material onwards. From the business perspective, the question about DRM is not about the technology itself: it is more about making the customer satisfied and pay for the content usage. In [25], the idea behind Apple's iTunes DRM approach can be summarized as a principle to hide the bad and promote the good:

> "Don't annoy the customers any more than necessary. Make it easy to take desired, legal actions and difficult to take the illegal or undesired actions."

Usually when DRM approaches are discussed, the content protection technologies are presented as a synonym for DRM. For example, Wikipedia describes the DRM technologies as a domain-specific encryption solutions[21]. However, the implementation of the *DRM system* concerns a variety of different technologies in addition to content protection and encryption: A DRM system usually provides the following functionalities:

1. identify the end user and other parties who are allowed to process[22]
2. identify the content to be protected
3. define the contract, that is, to define allowed and restricted operations for the specific user
4. protect the content and deny unauthorized access
5. track and maintain the restrictions for the whole duration of the contract made between the content provider and the end user(s).

The **end user identification**[23] is presumption for ecommercial DRM system: the contract is made with content provider and (legal) personality.

By the **content identification**, we mean the identification of the object concerned and the technological means for tracking the identity of the object. Within totally closed and controlled system, tracking technologies are unnecessary. However, the material to be consumed is always available for attacks, at

[21] http://en.wikipedia.org/wiki/Digital_rights_management

[22] Instead of restricting the access, the DRM system may provide restriction for variety of functions. E.g. in book domain, restricted operations may involve reading, printing, copying, loaning, modification, annotating, clipping, reselling of the material

[23] To be precise, the person identification is unessential, the question is about identifying the user profile in the system. For example with the mobile phones and Kindle, the user identity is based on id token of the device or SIM card. The implicit assumption is that multiple users don't use same device (but a user may use multiple devices).

least at the point when it's presented to the user[24]. For tracking the material (leaked) outside the system, fingerprinting or watermarking technologies may be utilized. In fingerprinting approaches, a unique identifier is generated from the content itself. In watermarking, an identifier is hidden inside the content. The possibly leaked material are checked by inspecting the fingerprint or watermark and compared to the id in the provider's database.

The **contract definition** defines the operations that are allowed and restricted according to the contract made between the parties. In the simplest case, the contract is implicit: the end user is able to open the content and consuming the material with the provided access. In the complex cases, the contract defines a set of operations, their durations and limitations and each request have to be interpreted separately. As an example, a contract might define that a book is 1) allowed to read for next 2 months, 2) partly allowed to print, 2 pages for a day in maximum, 3) not allowed to lend for friends but 4) quotes can be provided in maximum of 10 rows in each. In order to take care of the complex definitions, we have to have the system that supports expressing the special conditions and follows the restriction the whole life-cycle of product.

The **content protection** means the ways to protect or to hide the content from unauthorized access according to the contract defined. Content protection techniques provide usually means for secure delivery and content encryption strategies. Typical scenario for securing the eReading material is depicted in the State of the Art report[24, p. 42].

5.2 Functional overview of a DRM system

The main entities defined in the contract of a DRM architecture are the *users*, *content* and *rights*: a user has specific rights for content usage. The relations of the entities is depicted in the figure 3. The entities are listed from functional system perspective: what entities we have to model in order to provide the content and the rights for the end user. The system itself, the content provider or the rights between different content providers are not included in the figure.

An example of functional overview of the DRM architecture is shown in figure 4. On top, *IP Creation Capture* functionalities manage the workflow for handling the *intellectual property (IP)* rights of existing material to the system and ensuring that the legislative issues and rights are correct. On the middle part, *IP Asset Management* have to take care of storing and retrieving the content, rights and users (*Repository*). The *Trading* issues are related to payment issues, licensing, sharing royalties for the rights holders and so on. The *IP Asset Usage* is responsible for restricting the unallowed usage from the user and providing the usage statistics for the provider. [15]

For a DRM system providing support for advertisement should include components for handling advertisement metadata (in *Repository*), payment issues in *Trading* and providing means for tracking the efficiency of advertisement (in *Asset usage*)).

5.3 Open Digital Rights Language (ODRL)

The Open Digital Rights Language (ODRL) is a XML-based language for expressing, providing and agreeing rights information of any kind of content[16]. It

[24]For example, the analog hole http://en.wikipedia.org/wiki/Analog_hole

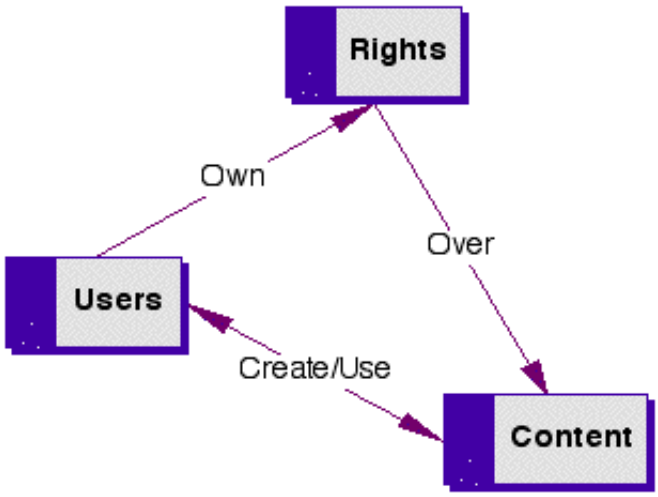

Figure 3: Entities of the DRM contract and their mutual relations [15]

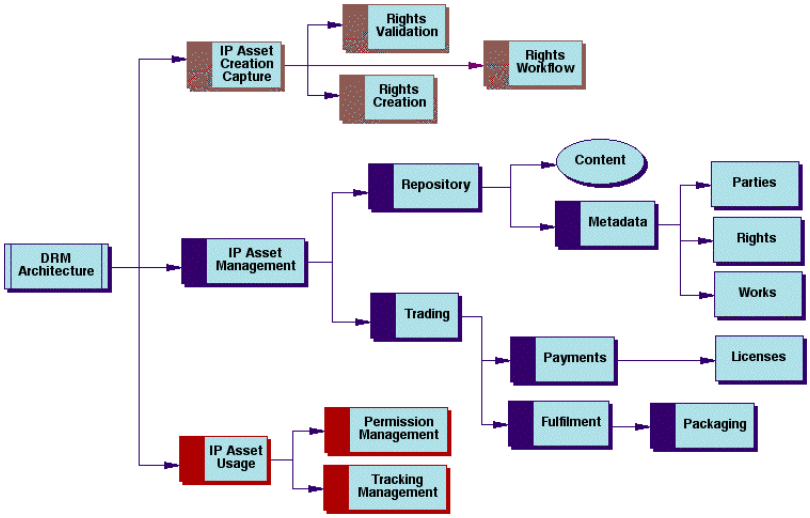

Figure 4: Functional overview of a DRM architecture [15]

doesn't define the framework or approach for content encryption and securing: the ODRL expression framework have to be provided inside *trusted environment* in order to control the usage of the content. The trusted environment can be achieved by 1) controlling both the client and server application and the communication between them, 2) having multiple parties that utilize common core components in unified way or 3) following common, legal agreements and practices in order to provide standard solution between different systems or applications.

The main entities of the ODRL language are *Party* (the users), *Rights* and *Asset* (the content). An item (e.g. a book) that is ready to be published (e.g. in a web store) is represented as an ODRL *offer*. When the user decides to buy

an object, an ODRL *agreement* is made between the provider and customer. Agreement defines the rights holder (the customer) and specific permissions for the content usage. In the figure 3 the ODRL components and their relations are represented.

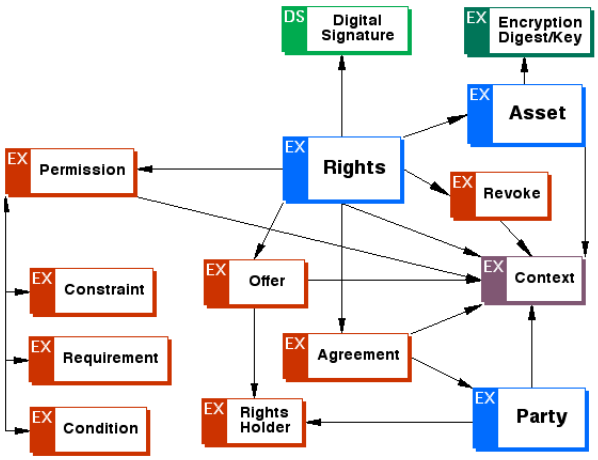

Figure 5: ODRL foundation model[16]

ODRL does not define exactly the nature or the way the *Party* and *Asset* element are defined. This is possibly because of the ODRL has to be implemented in a specific system and the representation of the content may vary in different systems. The only demand for those objects is that they have to have identifiers (URIs) [16]. In practice, the system have to provide also the resolution mechnism to access the actual content with the URI. Basically is is possible to define permission rights for any kind of content or it's subparts that whenever they have a unique identifier.

From the content provider perspective, a basic scenario of an ODRL based web store could be the following:

1. **Define the offering** Set the price and rights for the content to be provided.

2. **Publish the content** Provide the content to be available in the delivery system. Based on the *offering* metadata, the content price and usage rights are shown.

3. **Sell the content** User buys the content. When the money transaction is confirmed by the third party, the *agreement* is made between the content provider and customer.

4. **Deliver the content** Content is delivered for the customers client application using secured connection[25].

5. **Rights following** The client application will interpret the usage rights defined by the agreement and allow the corresponding functionalities. Temporary rights will be followed and disabled when expired.

[25]like in [24, p. 42]

6. **Statistics** Within the controlled, trusted application framework, the usage statistics can be collected.

5.4 ODRL Rights expressions

The rights in the ODRL language are defined by four kind of elements. (1) *Permissions* are specific actions (e.g. to display, to lend) that may have (2) *consraint* limitations (e.g. display for five minutes). (3) *Requirements* are conditions defined for a permission that have to be fulfilled in order to get the permission (e.g. display for five minutes by paying one euro). The last type of elements for rights expressions is (4) *condition*. Conditions define triggers for expiring the material. For example, if the client is not using authored device, then the material should be expired. In addition, conditions could be related on geospatial location or a time period. However, it should be noted that inspection of the fact that are the certain kind of conditions met, is strongly depending on the 1) ability of underlying device to provide the information and 2) the software capability to get that information from the device.

In figure 6, the action types of the permission model are represented. From the perspective of personal eReading devices, the pertinent action types are:

- **Display** Permission to read and view the content.
- **Print** Permission to print the content.
- **Annotate** Augmenting the content with personal notes.
- **Excerpt** Scrapbooking the content, making own archive of the content.
- **Sell, Lend, Give, Lease** Permissions for redelivering the content.
- **Save** exporting a copy of the item outside the system. E.g. creating a PDF version of the content from the viewer application.

With the permission constraints, the content provider can define restrictions on the content usage.

5.5 Summary of ODRL language

ODRL is a promising approach to formalize general model for defining formally agreements on content usage. With the open model, it is possible to inspect transparently different kind of approaches for implementing a DRM-based system. The ODRL specification acts as a good starting point on estimating the benefits and limitations of commercial solutions provided by the major players, such as Amazon (Kindle, Amazon store), Apple (IPad, Appstore) and Adobe (Digital Editions, AdobeContent Server).

A drawback of the ODRL language is the vagueness of the specification: the model does not provide detailed information *how* a specific kind of entity or expression should be interpreted. It is also likely that in a specific ODRL implementation, only a certain subset of functionalities are taken care within the language itself. For example, it is an open question should the revenue share between different authors be expressed inside the language or in other parts of the system.

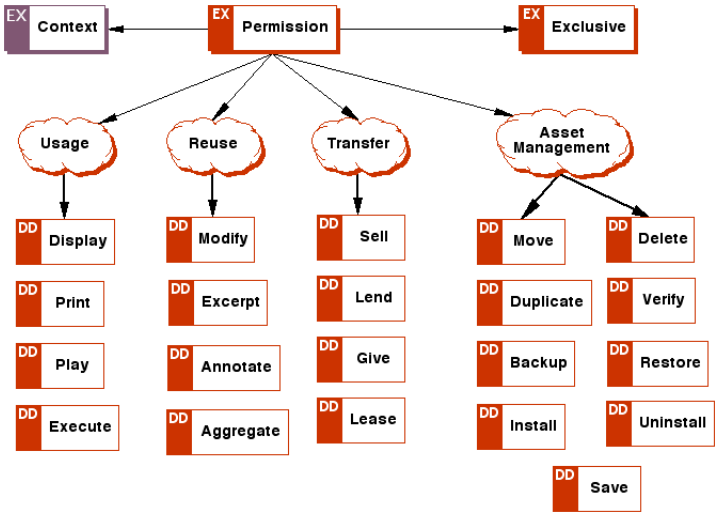

Figure 6: ODRL Permission Model[16]

The most notable system utilizing ODRL is OMA DRM[26]. Other standardization organizations involved in ODRL specification are W3C, NISO (U.S. National Information Standard Organization) and DCMI (Dublin Core Metadata Initiative)[27].

The new version of the ODRL is currently under development[28]. From the view of eReading domain, the main goals of the second version are (derived from [3]):

- **Downstream rights** Defining rights on several levels, including several participants: content creators, distributors and consumers. Consumers can be further divided to primary (the buyer) and secondary customers (the friends and related community).
- **Support for aggregated content** Better support for content packages and their rearrangements. Supporting specific rights for subparts of an aggregated content.
- **Support for status information** Support for modeling current status of the right. E.g. how many days or reading tiemes there are left before the content expires.
- **Semantics for contract negotiation** Support for negotiating prices and dynamically changing the price. E.g. providing cheaper prices for loyal customers, machine-assisted adjustment of prices based on the current market situation.

[26] *Open Mobile Alliance*, an organization developing standards for mobile industry. OMA DRM is utilized widely in the field with major players, such as Nokia, Ericsson, Siemens, Microsoft, IBM, Sun. For further information see http://en.wikipedia.org/wiki/OMA_DRM.
[27] Other organizations and companies can be found from http://odrl.net/news.html
[28] http://odrl.net/

- **Support for additional parties** Mechanisms for including additional parties in the contract. E.g. buying a book to other person as a gift, anonymous users with e-coupons.
- **Providing fair use mechanisms** Fair use exceptions of the DRM definition.
- **Extended expressivity** *NOT* operator (everything is allowed, except for XX), defining exclusive parties, defining rights for group of users.

Along with the changes to data modelling, ODRL v.2.0 should provide (more) unambiguous formal semantics, better support for existing metadata standards and strong emphasis on simplicity of the model[3].

5.6 Alternatives for ODRL

OMADRM, XrML, Mpeg-21, rights definitions inside other languages.

6 Additional models for eReading

In this chapter we list some relevant models that are not included in the report. The further inspection of the following models in the document is under consideration.

6.1 Prism

Prism (Publishing Requirements for Industry Standard Metadata) is an industry standard metadata specification for managing the content mainly in business-to-business scenarios. According to Prism homepage[29]:

> "Prism specification defines [...] vocabulary for managing, aggregating, post-processing, multi-purposing and aggregating magazine, news, catalog, book, and mainstream journal content"

Related to content description, PRISM FAQ[30] states that: "PRISM describes many components of print, online, mobile, and multimedia content including the following:"

- **Who** created, contributed to, and owns the rights to the content?
- **What** locations, organizations, topics, people, and/or events it covers, the media it contains, and under what conditions it may be reproduced?
- **When** it was published (cover date, post date, volume, number), withdrawn?
- **Where** it can be republished and the original platform on which it appeared?
- **How** it can be reused?

[29] Excerpt from Prism homepage http://www.prismstandard.org/about/
[30] http://www.prismstandard.org/faq/

6.2 Onix

Onix for Books is an book industry standard for representing and delivering book information in electronic format. It is maintained by EDItEUR[31].

6.3 HTML5

HTML5 is latest revision of the standard representation and content format for the WWW. The models utilizing HTML for the content representation (e.g. EPub format) are likely to be shifted in to HTML5 format in the near future. The key features of the HTML5 should be carefully inspected to estimate their impact on existing eReading solutions. At least HTML5 will provide native support for rich-media applications (similar to Flash) and better layout support, including support for multi-column pages. The specificatiob of the HTML5 is still a draft, current latest version of the draft is dated on 27th of August 2010[32].

A good introduction for the HTML5 features can be found from the web page and corresponding book Dive Into HTML5 by Mark Pilgrim.

6.4 AdsML

AdsML is a metalanguage and framework for managing advertisement solutions in eCommerce business[33].

7 Further reading

7.1 News

NewsML

- NewsML 1.2 manual
- NewsML G2 2.4 manual, including related G2 models
- NewsML for Dummies
- NewsML Semantic Web Project: main page, demo

Newsfeed languages

- Atom syndication format memo
- RSS 2.0 specification
- Comparison of RSS2.0 and Atom 1.0

[31] http://www.editeur.org/
[32] http://dev.w3.org/html5/spec/Overview.html
[33] see http://www.adsml.net/

7.2 Books

EPUB

- Open Publication Structure (OPS) v.2.0.1 draft
- Open Packaging Format (OPF) v.2.0.1 draft
- OEBPS Container Format (OCF) v.2.0.1 draft
- EPUB 2.1 Working Group Charter - Draft 0.11 5/7/10
- NewsML Semantic Web Project: main page, demo

PDF Adobe PDF reference v.1.7

7.3 Advertisement

- AdsML Framework home
- AdsML Framework overview
- AdsML Quick Start guide

7.4 DRM-related languages

- Adobe's XMP-language for embedding metadata inside files
- Embedding Creative Commons licence with XMP

7.5 Other related metadata standards

- Extensible Metadata Platform (XMP) specification. Language and platform for embedding metadata inside different file formats

References

[1] W3C Semantic Web Activity, 2001.

[2] Understanding Metadata, 2004.

[3] Open Digital Rights Language (ODRL) Version 2 Requirements, February 2005.

[4] Dublin Core Metadata Element Set, Version 1.1, January 2008.

[5] RSS 2.0 Specification, March 2009. Version 2.0.11.

[6] EPUB 2.1 Working Group Charter - Draft 0.11 5/7/10, May 2010.

[7] Adobe Systems Incorporated. *XMP Specification*, September 2005.

[8] Adobe Systems Incorporated. *PDF Reference, 6th edition, version 1.7*, November 2006.

[9] Tiina Alanko, Asta Bäck, Sari Vainikainen, Chip Gylfe, and Janne Saarela. D1.1 Inventory of the Types of Metadata in Editorial Operations, August 2009. Cross Media project.

[10] H. Alvestrand. RFC3066: Tags for the Identification of Languages. *Internet RFCs*, 2001.

[11] O. Corcho. Ontology based document annotation: trends and open research problems. *International Journal of Metadata, Semantics and Ontologies*, 1(1):47–57, 2006.

[12] N. Freed and N. Borenstein. RFC2045: Multipurpose Internet Mail Extensions (mime) part one: Format of internet message bodies. *RFC Editor United States*, 1996.

[13] N. Freed and N. Borenstein. RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. *RFC Editor United States*, 1996.

[14] Kelvin Holland. *G2 Guide for Implementers*. International Press Telecommunications Council (IPTC), November 2009.

[15] R. Iannella. Digital rights management (DRM) architectures. *D-Lib Magazine*, 7(6), June 2001.

[16] Renato Iannella. *Open Digital Rights Language (ODRL) version 1.1*. Open Digital Rights Language (ODRL) Initiative, July 2002.

[17] International Digital Publishing Forum (IDPF). *OEBPS Container Format (OCF) v.2.0.1*, May 2010.

[18] International Digital Publishing Forum (IDPF). *Open Packaging Format (OPF) v.2.0.1*, May 2010.

[19] International Digital Publishing Forum (IDPF). *Open Publication Structure(OPS) v.2.0.1*, May 2010.

[20] O. Lassila and D. McGuinness. The role of frame-based representation on the semantic web. *Link*
"oping Electronic Articles in Computer and Information Science, 6(5):2001, 2001.

[21] Laurent Le Meur, Michael Steidl, and Jayson Lorenzen. *NewsML 1.2 Guidelines V 1.01*. International Press Telecommunications Council (IPTC), February 2008.

[22] National Information Standards Organization. *Specifications for the Digital Talking Book*, April 2005.

[23] M. Nottingham and R. Sayre. The Atom Syndication Format, December 2005. RFC4287.

[24] Olli Nurmi, Katri Grenman, Atte Kortekangas, and Hannele Antikainen. State of the Art for the eReading. Technical report, VTT Media Technologies, June 2010.

[25] J.M. Van Tassel. *Digital rights management: protecting and monetizing content*. Focal Press, 2006.