Multilingual Composite Document Management Framework For The Internet: An FRBR Approach

Jean-Marc Lecarpentier
GREYC - CNRS UMR 6072
Bd du Maréchal Juin
14032 Caen Cedex, France
jml@info.unicaen.fr

Cyril Bazin
GREYC - CNRS UMR 6072
Bd du Maréchal Juin
14032 Caen Cedex, France
cyril.bazin@info.unicaen.fr

Hervé Le Crosnier
GREYC - CNRS UMR 6072
Bd du Maréchal Juin
14032 Caen Cedex, France
herve@info.unicaen.fr

ABSTRACT

Most Web Content is nowadays published with Content Management Systems (CMS). As outlined in this paper, existing tools lack some functionalities to create and manage multilingual composite documents efficiently. In another domain, the International Federation of Library Associations and Institutions (IFLA) published the Functional Requirements for Bibliographic Records (FRBR) to lay the foundation for cataloguing documents and their various versions, translations and formats, setting the focus on the intellectual work.
Using the FRBR concepts as guidelines, we introduce a tree-based model to describe relations between a digital document's various versions, translations and formats. Content negotiation and relationships between documents at the highest level of the tree allow composite documents to be rendered according to a user's preferences (e.g. language, user agent...). The proposed model has been implemented and validated within the Sydonie framework, a research and industrial project. Sydonie implements our model in a CMS-like tool to imagine new ways to create, edit and publish multilingual composite documents.

Categories and Subject Descriptors

H.3.2 [**Information Systems**]: Information Storage; H.5.4 [**Information Systems**]: Hypertext/Hypermedia—*Architectures*; I.7.2 [**Document and Text Processing**]: Document Preparation—*Format and notation, Hypertext/Hypermedia, Standards*; J.7 [**Computers in Other Systems**]: Publishing

General Terms

Design, Documentation

Keywords

Multilingual documents, Composite documents, Document Management System

DocEng2010, September 21–24, 2010, Manchester, United Kingdom.

1. INTRODUCTION

Most Internet documents nowadays are managed using Content Management System (CMS) such as blogs, wikis, *etc.* Usually, document content can be written in any language, then translated to another language in order to improve knowledge dissemination. Managing different versions of the same intellectual work (e.g. original version and its translations; its various formats) is a challenge for CMS. To be truly multilingual, relations between documents such as the inclusion of a document within another is a more challenging issue. For example, consider a French translation of an English text document including a diagram. When using the French version, the system has to display the diagram, its caption and metadata in French. A fallback mechanism displays the diagram's English version if it is not translated yet.

In this article, we introduce a model for managing such multilingual composite documents efficiently. Our approach considers the different versions of an intellectual work into a single tree-structured document, by analogy with libraries' FRBR approach. Thus, the different versions of the same intellectual work are aware of each other and we can manage relations between documents at the intellectual work level.

The remainder of this paper is structured as follows. Section 2 provides a review of two popular open source CMS, Section 3 introduces the Functional Requirements for Bibliographic Records. Section 4 presents how we adapt the principles of FRBR to create an original document model. Finally, Section 5 gives an overview of perspectives for future work.

2. WEB INDUSTRY OPEN SOURCE CMS

Wordpress [5] supports multilinguality through the use of plugins (we used WPML [6]). Blog posts can be translated and a post displays links to its available translations. A default language is defined for the site and each post must exist first in this default language before it can be translated. Media library documents (images, videos) have a title, caption, alternative text and description only in a single language. Including a media component in a blog post can then result in a caption displayed in a different language than the post itself. When the media is inserted, the media's information is copied into the post's content. The information is not up to date if the media's metadata change later on (for example caption or description). Because Wordpress inserts the media directly into the post content, no reverse mechanism can locate which posts use a given media.

The CMS displays only posts available in the chosen in-

terface language. Consequently, the latest posts appear on all versions of the site only if their translation is published. The user can not choose to have a complete multilingual list of new posts, even if not translated yet.

Drupal [1] is one of the most popular open source CMS. Drupal 7 (alpha version) handles multilinguality out of the box, but needs additional modules to make it user-friendly. A basic Drupal installation handles images in an article as mere files, eventually with the addition of alternate text for HTML rendering purpose. In Drupal 6, each translation of an article is a separate content node. Used in conjunction with the CCK module, linking a page to an article would then refer to several nodes, even though it is the same intellectual content. An extra function is required to *group* the translation nodes.

In both cases, an image included in an article will only display very limited metadata such as title, caption, alternative text and filename. Image indexing is reduced to techniques that must *guess* the information associated to the image, either by analyzing the image itself or the content around.

CMS multilingual support is usually developed as an add-on to an already existing system. Moreover, relationships between a *master* document and its components is usually implemented as a mere relationship between a document and a file. Let us take for example an article, including a technical diagram. When translating the article, a CMS will not be able to provide the translator with the other available versions of the diagram (if any). Each image file is independent of the others, whereas it should be able to know that there is another image that is the "translation" of the diagram.

These identified weaknesses in content management systems are relevant to the underlying model of a document. They are the starting point of our research.

3. FUNCTIONAL REQUIREMENTS FOR BIBLIOGRAPHIC RECORDS

In order to base cataloguing rules on the intellectual work instead of the physical items on a shelf, the International Federation of Library Associations and Institutions (IFLA [1]) has published the Functional Requirements for Bibliographic Records (FRBR) [10]. FRBR defines a model for bibliographic records based on entity relationships. This model has also been adopted by the International Committee for Documentation (CIDOC) of the International Council of Museums (ICOM) with an object oriented model.

FRBR defines entities that represent the key objects of a document. These entities are organized in three groups. The first group defines *Work*, *Expression*, *Manifestation* and *Item* and represents the *model* of a set documents. The second group defines the entities *person* and *corporate body*, which represent entities responsible for the editing process of a document (creation, production, publication, etc.). Finally, the third group defines entities that are the subjects of a document (*concept*, *object*, *event*, and *place*).

The first group's four hierarchical entity levels *Work*, *Expression*, *Manifestation* and *Item* can be illustrated as shown in Figure 1 and are defined in the FRBR report as follows:

> "The entities defined as *work* (a distinct intellectual or artistic creation) and *expression* (the intellectual or artistic realization of a work) reflect intellectual or artistic content. The entities defined as *manifestation* (the physical embodiment of an expression of a work) and *item* (a single example of a manifestation), on the other hand, reflect physical form."

[1]IFLA, http://www.ifla.org

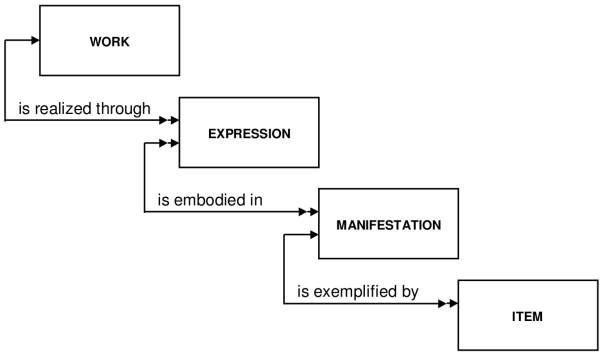

Figure 1: Group 1 entities (from FRBR report)

A tree structure represents any intellectual production. Multilingual support is achieved through the Expression entities, and different formats through Manifestation entities.

FRBR also defines relationships between entities to express links between them and are used "as the means of assisting the user to *navigate* the universe that is represented". The *has part* or *is a part of* Work-to-Work relationship can be used to store the relationships between a *master* document and its components, therefore keeping that information at the highest level of the document tree. The *has a translation* or *is a translation* Expression-to-Expression relationship can obviously be used to store the fact that an expression is a translation of another one.

We believe that the FRBR model reaches beyond documentation and its principle can be a good starting point for modeling and managing composite digital documents.

4. A DOCUMENT MANAGEMENT FRAMEWORK BASED ON FRBR

FRBR was designed for libraries with physical items on shelves. When dealing with a digital library or a CMS, the managed documents are obviously not on a shelf. Moreover library catalogs only store documents' metadata, whereas a CMS actually stores a document's metadata and its content. Therefore the FRBR model serves as guidelines and its principles need to be adapted to convey a CMS' specific needs.

The *Work* entity is the intellectual creation, therefore it contains only metadata such as the first publication date, and additional metadata depending on the type of document dealt with. Expression entities represent the various translations of a Work. A "reference expression" refers to the original expression. A Manifestation entity is an embodiment of an expression of a work. Manifestation entities can refer to various formats, such as HTML, PDF or various media encoding. A "reference Manifestation" refers to the occurrence that served for the creation of the other ones. Figure 2 illustrates the resulting tree.

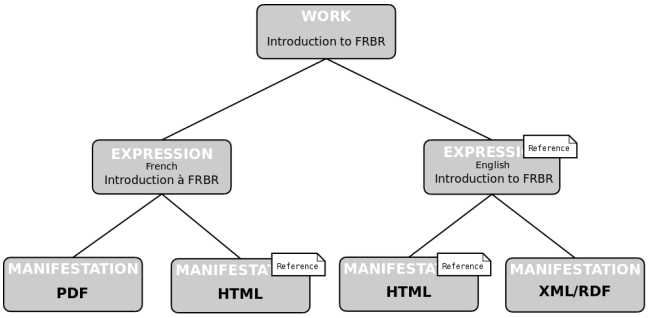

Figure 2: Document tree showing various translations and formats

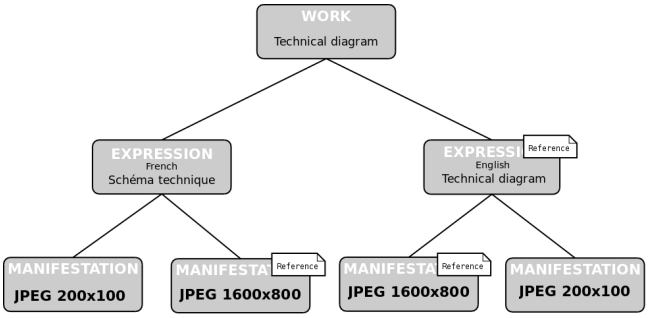

Figure 3: Image document tree showing various translations and sizes

In the example of an image, the Expression entities contain the translations of the caption and description. Manifestations refer to various image sizes needed to display the image in different situations (thumbnail, square, web, original size, etc). If the image itself is translated, for example an illustration containing text and labels that are translated too, then Expressions and Manifestations refer to different files, resulting in a tree as shown Figure 3.

Using such a representation, many files can be grouped in one document tree which represents the relations between them. Moreover, with this model, each node can have specific metadata.

Through its entity levels, FRBR defines a tree structure, and our model for documents uses the same principle: a document is a tree where the nodes are FRBR-like entities. Each entity is linked to specific data or metadata via statements, using a RDF-like subject-predicate-object structure. Each document class defines its own set of acceptable specific data, making it simple to create different types of documents (News, Article, Image, E-book, etc.).

With this model, a document is self aware of its many available versions, for instance the different translations and formats. Since a composite document groups many files, the system has to choose which branch of the document tree it must use and display, in order to best match the user's preferences. Each entity of a document tree has its own identifier. Therefore a user will request an entity, and depending on the entity level requested (Work, Expression or Manifestation), a content negotiation will occur.

As outlined in *Cool URIs for the Semantic Web* [7], a W3C note published in December 2008, HTTP Language and Content Negotiation can be used to serve the most suitable corresponding content to the client's preferences. Keeping in mind that a client will always be served a Manifestation, providing an access to a document through the Work or Expression entity level follows the algorithm:

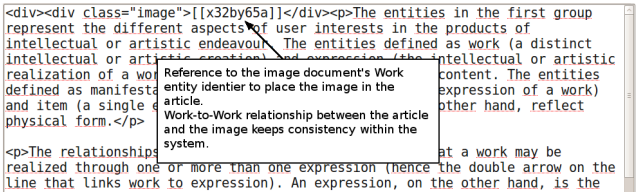

Figure 4: Edition of the content of article

1. at the Work level, Language-Negotiation is used to know which Expression of the document to use.

2. at the Expression level, Content Negotiation is used to decide which Manifestation to serve. A typical use case would be a web browser accessing a resource and being served HTML content whereas a robot would be served an XML or RDF content.

3. at the Manifestation level, the system can directly serve the content.

Language and Content Negotiation also occurs in the system itself when dealing with composite documents. Let us take the example of an article containing an image. The image itself is a document with its own associated tree. *has part* and *is part of* relations between the image's Work entity and the article's Work entity define that this image document is a part of the main article. It is important to note that to edit the article, it is only needed to add a reference to the Work entity of the included image to define *where* the image representation will be displayed (shown Figure 4). The Work-to-Work relationship and Content Negotiation do the rest of the work. Without a WYSIWIG editor yet, the insertion of a component is made with a wiki-like syntax (e.g. `[[componentId]]`) to ensure ease-of-use by non technical users. Components thus must be stored within our system. With more complex editing processes and insertion of external components, an XLink syntax may be used.

The system uses Content Negotiation to find which Manifestation of the image to place in the article. For example if an Expression of the article in French is to be served, the Work-to-Work relationship will allow the system to find a French expression of the image. The article will include the image's French caption, description and metadata. If a French Expression for the image document does not exist, then the reference Expression is used as a fallback. The same negotiation occurs at the Manifestation level: if the image with the default size for article is not found, then the reference Manifestation can be used.

Since the image itself is a document with metadata, it provides search engines with informations such as keywords, author, copyright, improving indexing and discovery.

The *has part* and *is part of* Work-to-Work relations set between the article and the image solve some of the weaknesses outlined in 2. An image metadata, caption, etc. is consistent in all articles. The relations provide a reverse mechanism to know where the image is used as an illustra-

tion. Finally, each part of a document can display its own metadata.

5. PERSPECTIVES AND FURTHER WORK

In this paper we introduce a new model for managing multilingual composite documents. Relationships between documents allow inclusion of a document within another, enable consistency in the documents' data and provide metadata information that is usually not available. Content negotiation allows a user to be served the most suitable version of a document according to his/her preferences, as well as keeping consistency when a document in included into another. For indexing documents, we developed a simple tag system, where tags are attached to Work entity of a document, allowing for a basic multilingual tag support.

We implemented this approach, creating the R&D Document Management System named Sydonie (SYstème de Documents Numériques pour l'Internet et l'Edition). Sydonie's goal is to imagine new ways to create, edit and publish multilingual composite documents. Sydonie is funded by the Région Basse-Normandie at the University of Caen and sponsored by TGE-Adonis, a CNRS project aimed at Humanities researchers. It is developed in conjunction with C&Feditions, a publishing partner developing e-books services based on Sydonie.

A more in-depth evaluation of the system needs to be performed, especially with documents containing components such as video, audio, animations, boxes, etc. A potential drawback to address is the asynchronous evolution of the different versions of a document. A reference Expression can be updated, causing some translations to be out of date with the original version. To overcome this issue, an alert system to inform readers and authors about any such changes is to be designed. As of today, Sydonie is aimed at information professionals, aware of editing processes and translation problems, making this issue less critical than for an open system such as a wiki for example.

As a work in progress, Sydonie needs more features. The current model is being extended in order to provide more Expression-to-Expression relationships, for example *is an adaptation of*, *is a summary of* as expressed in the FRBR report. FRBR's group 2 and 3 entities are also being adapted to express relations to subject and authorship. Extensive metadata management (Dublin Core [2] support, XMP [3] metadata, *etc*) and synchronization between files' metadata and the document tree's metadata.

Further work needs to improve multilingual management of subject entities, using SKOS [9] to create multilingual *thesauri*. Being an open source project, it is important to achieve interoperability for document import/export. TEI [4] and PRISM [8] are currently being studied to choose the most suitable to our needs.

The FRBR Item entity used to store the information about *physical* items with a specific spot on a library's shelves could be thought of a way to store the various URIs of a document. Different locations of a Manifestation could be listed, for example with mirror servers (serving exactly the same pages), with a different *page environment* (i.e. the many informations *around* the document vary depending on the context of the displayed page), or with other locations when a same content is published by third party websites.

6. REFERENCES

[1] Drupal open source cms, 2010. `http://drupal.org/`.

[2] Dublin core metadata initiative, 2010. `http://dublincore.org/`.

[3] Extensible metadata platform, 2010. `http://www.adobe.com/products/xmp/`.

[4] Text encoding initiative, 2010. `http://www.tei-c.org/`.

[5] Wordpress open source cms, 2010. `http://wordpress.org`.

[6] Wpml wordpress plugin, 2010. `http://wpml.org/`.

[7] W. I. Group. Cool uris for the semantic web. Technical report, W3C, December 2008. Available at `http://www.w3.org/TR/cooluris/`.

[8] IDEAlliance. Prism: Publishing requirements for industry standard metadata. Technical report, International Digital Enterprise Alliance, Inc., 2009. Available at `http://www.prismstandard.org/`.

[9] A. Miles and S. Bechhofer. Skos simple knowledge organization system reference. Technical report, W3C, August 2009. `http://www.w3.org/TR/skos-reference/`.

[10] I. S. G. on the Functional Requirements for Bibliographic Records. *Functional Requirements for Bibliographic Records*. K. G. Saur, München, Germany, 1998.