Report on the Development of Breadcrumbs Navigation Feature in EPrints

Tomasz Neugebauer (tomasz.neugebauer@concordia.ca) and Bin Han

Concordia University

Montreal, Quebec, Canada.

Open Repositories 2010, in Madrid, Spain.

July, 2010

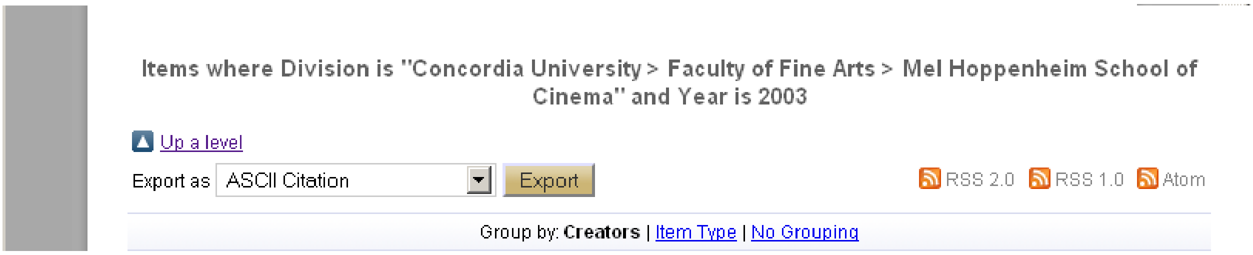
1. Introduction

Bread crumb navigation is a widely used and increasingly popular and technique for improving usability[1]. It uses a single line of text to show the current webpage's location in the site hierarchy and offers one-click access to higher site levels.

EPrints[2] system is a flexible platform for building a digital repository of research outputs of literature, scientific data, theses and reports or multimedia artifacts from collections, exhibitions and performances.

Spectrum[3]*,* Concordia's research repository is an instance of EPrints. The items in Spectrum have been categorized in browse views by the year created, department, author, and document type. For each category, we have further labels, for example, authors are indexed alphabetically. The organization structure is shown in Figure 3.

The default breadcrumbs in EPrints are composed of text without hyperlinks. For example, the following (Figure 1) is the breadcrumb that appears while navigating to the year 2003 publications of the Cinema department:

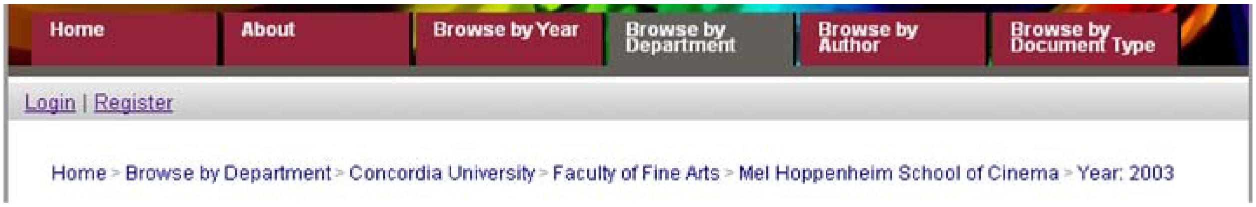


Items where Division is "Concordia University > Faculty of Fine Arts > Mel Hoppenheim School of Cinema" and Year is 2003

Export as ASCII Citation

Figure 1 Default Breadcrumbs

We have written a breadcrumb function, called from *dynamic_template.pl* that transforms the above textual breadcrumb into the conventional breadcrumbs that allow once-click access to higher site levels (as shown in Figure 2). In the following sections, we introduce the implementation detail.



Home > Browse by Department > Concordia University > Faculty of Fine Arts > Mel Hoppenheim School of Cinema > Year: 2003

Figure 2 New Breadcrumbs

[1] J. Nielsen. Breadcrumb navigation increasingly useful. Alertbox, April 2007. <http://www.useit.com/alertbox/breadcrumbs.html>

[2] EPrints - Digital Repository Software. University of Southampton, UK. 2010. <http://www.eprints.org/software>

[3] Spectrum: Concordia University Research Repository. Concordia University. 2010. <http://spectrum.library.concordia.ca>

2. Related concepts in EPrints APIs

The implementation of breadcrumbs uses the API provided by the EPrints system, so it is necessary to cover some of the API calls that have been used.

2.1) Uri

Uniform Resource Identifier (URI) is a string or characters used to identify the resource on the Internet. The breadcrumbs are generated based on the URI of each page. With EPrints API , *$session->get_request->uri*, we can easily get the URI.

2.2) Utils

As indicated by its name, *EPrints::Utiles* module provides basic utility methods. For example, *EPrints::Utils::escape_filename,* takes a value, such as an author name or subject id and escapes it into a portion of the corresponding EPrints URI. *EPrints::Utils::tree_to_utf8* converts an XML DOM tree that is returned by some functions and converts it to a utf-8 encoded string.

2.3) Subject

EPrints provides the capacity of customizing faculties/schools divisions' hierarchy as shown in Figure 3. In EPrints system, each division is a *subject* object which contains *subjectid*, *name*, and *parents* (the parent node in the division tree).

2.4) EPrint

EPrints::DataObj::EPrint represents a single deposited item and the metadata associated with it. As one of the core objects in the system, it is easy to get associated information such as the *title*, *author*, and etc. by *$eprint->get_value* once the eprint object is created.

3. Operations for each context

As mentioned in section 2.1, it is important to know that the breadcrumb is developed based on the URI of each page, so any changes to EPrints URI schema may affect the rendering of the breadcrumb.

By analyzing the URI of each page, we provide breadcrumbs of each category, *the static pages*, *the cgi pages*, *the view pages* and *the abstract pages*. The code for breadcrumbs function is defined in:

/opt/eprints3/archives/ARCHIVEID/cfg/cfg.d/bread_crumb.pl

Invoked from:

/opt/eprints3/archives/ ARCHIVEID/cfg/cfg.d/dynamic_template.pl

By:

$parts->{bread_crumb} = $session->get_repository->call("generate_bread_crumb", $session);

3.1) Static pages

Static pages stand for the pages that manually created with the default template in the directory: /opt/eprints3/archives/ARCHIVEID/cfg/lang/en/static , such as the faq page, the contact page, etc.

There are two hand code hash tables containing the custom breadcrumb information of the static pages:

- *%static_pages* is a hash table includes the URI of the static page as the key and its label for the breadcrumb as the value. For example, *'information.html' => "About",* results in the page *[ARCHIVE]/information.html* with the breadcrumb: Home > About.
- *%static_pages_structure* is a hash of array to maintain custom hierarchical information for static pages. For example, *'faq.html' => ['information.html', 'faq.html'],* results in the FAQ breadcrumb to be interpreted as hierarchically below "About" (information.html): Home > About > FAQ

With the above solution, when adding or modifying static page structure, we only need to modify the two hash tables without the need to modify the function that renders the breadcrumb (*sub render_staticpage_crumb*).

3.2) Cgi pages : Search and Latest Additions

We have applied breadcrumbs on two pages in the cgi directory (/opt/eprints3/cgi): search and latest additions. As in the case of static pages, a hash is created to maintain breadcrumb of these:

```
%cgi_pages = (
                'latest' => "Latest Additions",
                'search' => "Search",
    );
```

3.3) View pages

The URI of all the view pages use the following syntax: *http://spectrum.library.concordia.ca/view/**

We treat 4 common browse types:

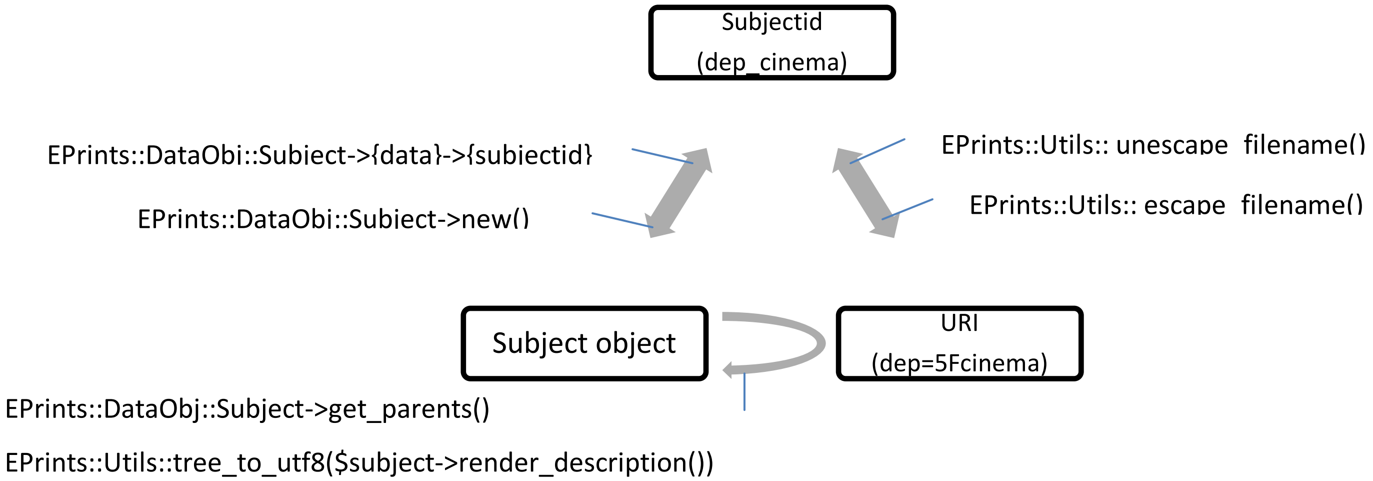
3.3.1) Browse by Year

When the URI matches: */view/year/(\d+).html* , the breadcrumb is produced as:

Home > Browse by Year > Year: (\d+)

3.3.2) Browse by Department

On the browse by department page, when the URI matches */view/divisions/subjectid/**, we need to work among three EPrints concepts: *URN*, *subjectid*, and *subject object*. The diagram below indicates their relationships and the EPrints calls.


Subjectid
(dep_cinema)
EPrints::DataObj::Subject->{data}->{subjectid}
EPrints::DataObj::Subject->new()
EPrints::Utils:: unescape_filename()
EPrints::Utils:: escape_filename()
Subject object
URI
(dep=5Fcinema)
EPrints::DataObj::Subject->get_parents()
EPrints::Utils::tree_to_utf8($subject->render_description())

From the URI (/view/divisions/dep=5Fcinema/), *dep=5Fcinema* is extracted and we can get the *subjectid* (dep_cinema). With the *subjectid*, the subject object can be created from which we get the name (Mel Hoppenheim School of Cinema) and parent *subjectid* of it. We apply this procedure recursively, until the root (university).

The table below is a concept view of subject object of 'Mel Hoppenheim School of Cinema'.

Subject Object		**EPrints APIs used**
subjectid	dep_cinema	$subject->{data}->{subjectid}
name	Mel Hoppenheim School of Cinema	EPrints::Utils::tree_to_utf8($subject->render_description())
parents (subjectid)	fac_finearts	$subject->get_parents()

Finally, the information of the URI is extracted and analyzed by regular expression for any further subdivisions by year and the final breadcrumb is produced shown in Figure 2.

3.3.3) Browse by Author

When the URI matches */view/creators/(.+)=(.+)=(.+)=* , the breadcrumbs is produced as:

Home > Browse by Author > Index Character > Author Name

The Index Character is derived by regular expression, and the 'clean' author name is returned from the *Utils* method unescape_filename, for example:

Aaron, Mandie = EPrints::Utils::unescape_filename(Aaron=3AMandie=3A=3A);

3.3.4) Browse by Document Type

When the URI matches */view/doctype/(.+).html* , the document type is extracted and the breadcrumbs is created with *unescape_filename* method.

3.4) Abstract pages

The URI of the abstract page (\d+) contains a series of numbers, which is the eprintid of the corresponding eprint object. We create the eprint object from the eprintid and get its parameters as needed.

The hierarchical position of the abstract pages is immediately below Home, since a publication may belong to more than one department. We decided to use the title of the Eprint as the final breadcrumb on the abstract pages.

Home > EPrint Title

4. The Configuration

Each page in the repository is using a template which is an XHTML file with some additional EPrints calls. A custom layout can be applied to the pages by editing this template directly.

/opt/eprints3/archives/ARCHIVEID/cfg/lang/en/templates/default.xml

There are number of *epc:pin* elements to tell EPrints how to combine the template with page-specific content to build each individual page.

4.1) Breadcrumb and title pins and files

The implementation files are available in Eprints Files [4].

In our application, we defined two new pins in the template file, the <epc:pin ref="bread_crumb"/> and <epc:pin ref="abstract_title"/>, one for the breadcrumb and the second to present the title information on the abstract page.

In the configuration file,

/opt/eprints3/archives/[ARCHIVEID]/cfg/cfg.d/dynamic_template.pl

We added two lines of code to reference the implementation of the two pins:

```
$parts->{bread_crumb} = $session->get_repository->call("generate_bread_crumb", $session);
$parts->{abstract_title} = $session->get_repository->call("generate_abstract_title", $session);
```

The two function 'generate_bread_crumb' and 'generate_abstract_title' are located in two separate files at:

/opt/eprints3/archives/[ARCHIVEID]/cfg/cfg.d/bread_crumb.pl
/opt/eprints3/archives/[ARCHIVEID]/cfg/cfg.d/abstract_title.pl

There is a default title for each page, implemented by <epc:pin ref="title"/> by EPrints system (see Figure 1). For the view pages, the 'title' pin is the static breadcrumb; for the abstract page, it is the title of the item; for the search page, it provides the search information. Since we have developed the new breadcrumb and also customized the title information for the abstract pages, it is necessary to turn off the default title pin in some cases, we decided to use Javascript for this. In

/opt/eprints3/archives/[ARCHIVEID]/cfg/static/javascript/auto/90_local.js

We developed a function to detect the URI of the page and determine which segment to turn off. Table 2 shows us the enabled pin on each page.

	static page	search page	view page	abstract page
'title' pin		√		
'bread_crumb' pin	√	√	√	√
'abstract_title' pin				√

Table 1 pin activation

4.2) CSS

We placed some additional CSS for the presentation of the breadcrumb in the file *zzz_local.css*

[4] Neugebauer, Tomasz and Bin, Han (2010) Breadcrumbs Navigation. EPrints Files. <http://files.eprints.org/555/>

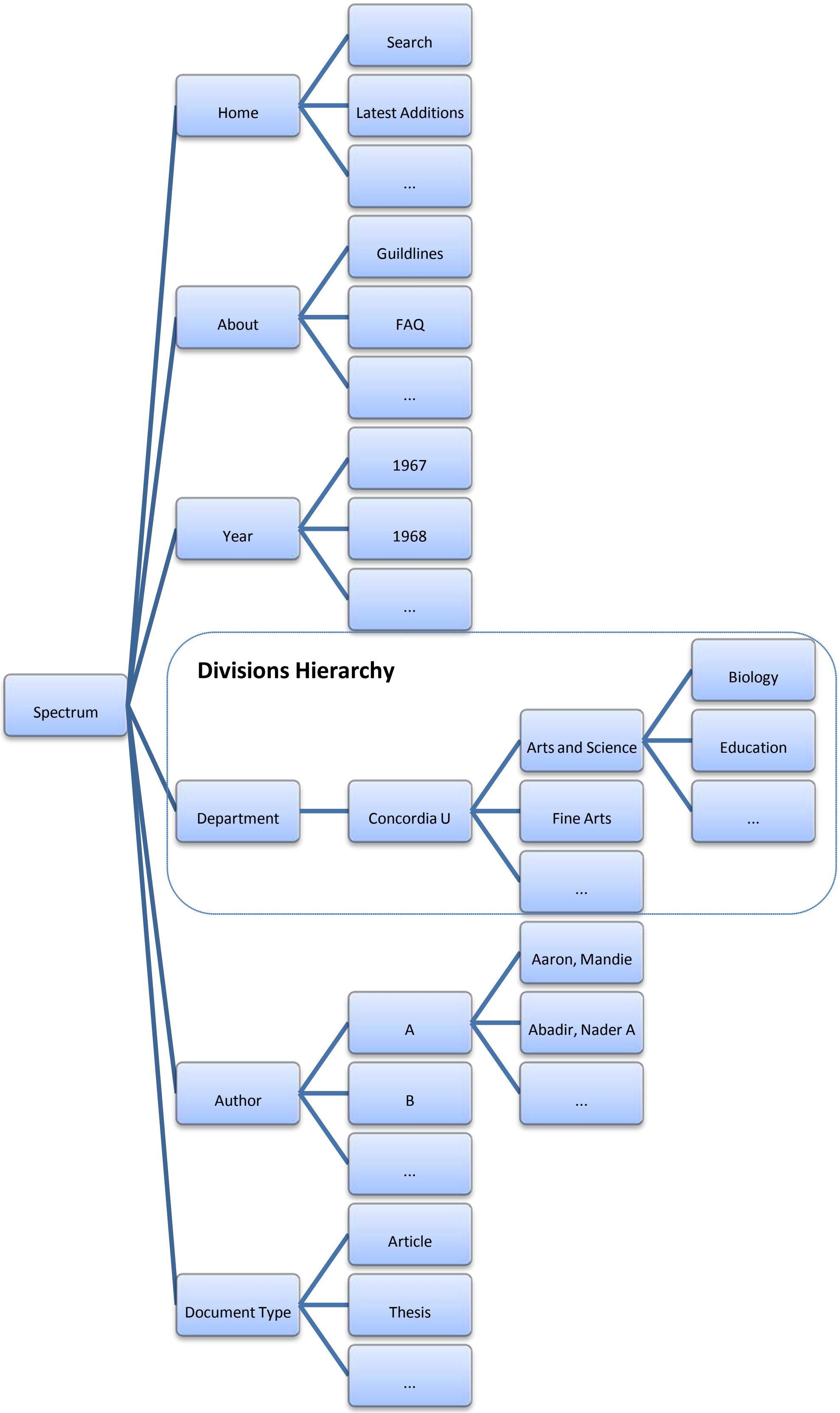


Figure 3 Spectrum Organization Structure