Integration of open source systems for visibility of scientific production of universities

Ionela Birsan, Daniela Drugus, Marius Stoianovici, Angela Repanovici

***Abstract*—** The article shows us the research conducted in order to find the barriers to dissemination and communication of scientific articles published by such institutional digital repositories. Transilvania University of Brasov, Romania, provides a new service attached to the digital repository, an automatic query interface of the SHERPA RoMEO platform, a publisher reviewing platform. Pressing the new button will open a new window (pop-up) of the browser, where the actual query of the SHERPA RoMEO server will be performed then, upon further closing the window, the user will automatically switch to the next page – the second step of submitting the new item (where account is taken of the options being checked in the first page, even if the newly created button was pressed instead of the "*Next*" button). The transformation results in a HTML file. It is a simple list of the identified publishers and the "*romeo*" color associated with each of them. The software application developed with very low cost price can be also used as a model for other universities. The application is original, the model is easy to develop, and the practical implications are of great use to the academic community**.**

***Keywords*—** Digital repository, DSpace, Romania, SHERPA RoMEO, XML, XSL.

I. Introduction

THE movement of open access to information has developed new models of communication and dissemination of scientific information. Universities have provided the academic community with instruments to promote scientific production, creating institutional digital repositories. The academic community reacted with disbelief, the number of archived documents being below expectations. Most authors are afraid of breaching copyright and do not want to devote time to reviewing the publishing conditions imposed by publishers.

Many surveys have been conducted on the academic community's behavior concerning the Open Access movement and the motivations and impediments for which they archive or not their papers published in open access.

The vast majority of surveys refer to the behavior of researchers on archiving in institutional digital repositories. [1], [2].

The main barriers encountered are: copyright concerns [1], [3], [4], [5], [6], additional time and effort [7] (Van House, 2003), mistrust [8].

Faculty with technical skills and younger faculty are more involved in self-archiving articles. Providing logistical and technical support will also foster participation of those who are less computer adepts [9],[10],[11],[12].

Issues relating to copyright and intellectual property are also generated by not knowing the conditions within the publishing agreements. There is misconception that self-archiving breaches copyright agreements [13].

Authors fear that, by the publishing agreements entered into, they are not allowed to upload their papers in institutional digital repositories. One in ten authors knows such problems, while the other nine have only "a slight idea". This ignorance leads to a tendency for authors to be over-cautious [14].

Copyright remains the biggest obstacle in self-archiving articles in institutional digital repositories [15], [16], [17].

II. Implementing and using the DSpace platform

DSpace [18] is a software application (platform) designed for academic, non-profit and also commercial organizations with a view to developing and managing digital repositories. The digital repository is a collection of digital documents, organized in a well-defined hierarchical structure. DSpace software is free of charge and easy to install (*out of the box*), and fully customizable, in order to suit any organization's needs.

DSpace preserves and enables easy and open access to all types of digital content including text, images, moving images (video) and data sets. With a growing community of developers committed to continue the software's expansion and improvement, each application installation benefits from the experience of the previous users and developers.

DSpace is the software support of a digital repository of documents. In turn, the digital repository is the environment (software) where the *institutional digital repository* may be created. There are several definitions of the "institutional repository". Lynch [19] defines the institutional repository as: "a set of services that a university provides to its community members, for the management and dissemination of digital materials created by the institution or by the members of such

Ionela Barsan is with Transilvania University of Brasov, Romania (e-mail: one.barsan@yahoo.com.)

Daniela Drugus is with University of Medicine, Iasi, Romania(e-mail: daniela.drugus@umfiasi.ro).

Marius Stoianovici is with Transilvania University of Brasov, Romania(e-mail: marius_stoianovici@yahoo.com)

Angela Repanovici is with the Product Design and Environment Department at Transilvania University of Brasov,Romania(e-mail: arepanovici@unitbv.ro, telefon 40745820361, corresponding author).

community. Organizational commitment is essential to manage such digital materials, including long-term preservation where appropriate, as well as organization, access or distribution". Ware [20] also includes the participation of the open archives initiative (OAI - Open Archives Initiative [21]): "a web-based database (repository) of scholarly material which is institutionally defined (as opposed to a subject-based repository); cumulative and perpetual (a collection of record); open and interoperable (using OAI-compliant software); collecting, disseminating and storing (is part of the process of scholarly communication). In addition, most would include long-term preservation of digital materials as a key function of the institutional repository".

For DSpace to become an institutional repository, special attention should be given to configuring and managing the same. But DSpace can be the support of any other type of digital repository, the "institutional" or "subject-focused" or any other character of the repository being brought about by the way the software application is further configured and managed.

Almost 1,500 installs of the DSpace software platform are currently known worldwide, most of them in the academic environment [22]. Out of these, 9 are in Romania, of which 6 are in the academic environment (one of them being the *Aspeckt* platform of Transilvania University) [22],[23]

III. SHERPA RoMEO PROGRAMMING INTERFACE

RoMEO is a database of publisher copyright policies on self-archiving, based on the publisher's copyright transfer agreement. It is maintained by SHERPA with support by JISC and the Wellcome Trust. Individual journal titles, ISSNs or publishers can be searched, and each title is identified as Green (can archive pre-print and post-print), Blue (can archive post-print (i.e. final draft post-refereeing), Yellow (can archive pre-print (i.e. pre-refereeing), or White (archiving not formally supported).

API (*Application Programming Interface*) is an acronym used to generally refer to collections of predefined software functions that allow writing custom applications running in a predefined environment. The application programming interface for SHERPA RoMEO is a machine/machine interface allowing programmers' access to Sherpa Romeo data within their own developed applications. For example, API can be used to embed automated searches of journals or publishing houses during a record (submission) process of a paper in a repository.

Like most APIs used in the web environment, the SHERPA RoMEO interface does not involve downloading a library of functions on the user's computer, but calling functions from a web application server, by HTTP queries.

IV. INTEGRATING SHERPA/ROMEO WITH DSPACE

Integrating information contained in the Sherpa/Romeo database into DSpace platform can be made by combining all the information presented in the previous chapters. A general diagram of the process is shown in Fig. 1.

There are two versions of the XSL language used in practice:

- XSLT [24], defining the transformations being applied to the XML tree;
- XSL-FO (*Formatting Objects*), used to transform XML documents into binary format documents such as PDF or even Microsoft Word.

There are three ways that an XML document may be transformed into another type of document by applying an XSLT stylesheet:

- the XML document and the associated stylesheet are sent to the client application (browser) whose task is to effectively perform the transformation according to the information in the XSLT stylesheet. In such conditions, server load decreases but the browser should allow processing of XML documents;

- applying the XSLT stylesheet is carried out on the server itself, the resulting document (usually in HTML format) being sent to the client. Thus, processing may be carried out according to the nature of the client program;

- the third possibility is very rarely used and refers to the transformation of the XML document using an external application and placing the resulting document (HTML) on the server, being further sent to the client.

The core element of the XSLT technology is the template: <*xsl:template*>. Two important elements may be found herein: the *match* attribute – specifies a path to the input tree; the content – implements the way transformation is performed.

The general form of a template is:

```
<xsl:template match="element_XPath">

...

<xsl:template>
```

The association of an XML document with an XSLT stylesheet is performed within the XML document by the processing instruction <*?xml-stylesheet>:*

```
<?xml-stylesheet href=”stylesheet/Login”
type=”text/xsl” />
```

The *href* argument specifies the name of the XSLT stylesheet and, where appropriate, the path thereto.

Identifying the fields within the tree structure of the XML document is performed through *XPath* elements (which is sometimes described as a language, although it is not a language proper). The XPath convention is similar in functionality to navigation through the directory structure in the operating systems, such as MsDOS, Linux or Windows.

At conceptual level, at the basis of the XML document's structure (but having no corresponding element within XML elements), is the root of the document, represented by the "/" character.

XPath expressions are interpreted from left to right, for example, for an XML tag which is at the first level of the tree structure (for example <*basis*>), the expression that reads the element value is "*/basis*", and for the following levels, it could be, for example "*/ basis/level1/level2*" and so on. The previous expression can be understood as: "starting from the document root, select the <*basis*> element, which is its child (the root's)." Failing to write the "/" character in the previous XPath expression radically changes the meaning of such expression, in which case "all the <*basis*> elements, which are children of the current node are selected". In the case of more complex XPath elements, the constituents are separated by the "/" character, which, as can be noticed, has a double meaning depending on the position in which it appears within the XPath element.

Moreover, there are certain situations where an XPath element must make a much more rigorous selection of the elements selected and treated within a template. It can be assumed, for example, that in a certain context only selecting the <input> elements is wanted, whose "type" attribute has a value different from "hidden". In order to achieve this, the element which will be filtered must be followed by the filter to be applied. It consists of a pair of brackets ([]), which usually frames a condition.

While defining a new submission for a collection on the DSpace platform, a query is sent to the Sherpa/Romeo server, using the application programming interface it provides, which responds by an XML document containing the information required. The XML document is then processed through the XSLT transformation so as to generate the content displayed on the HTML page.

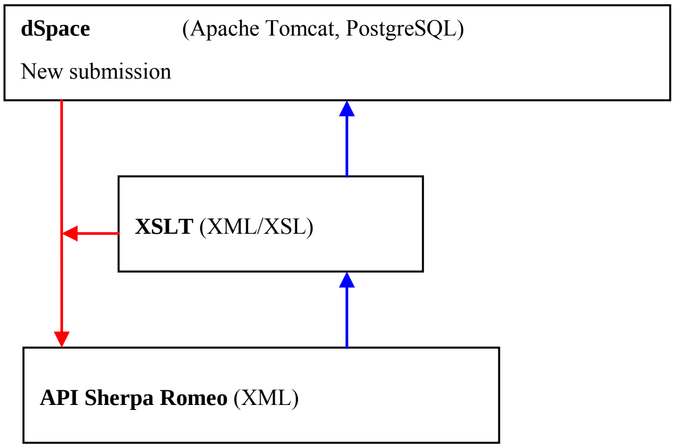


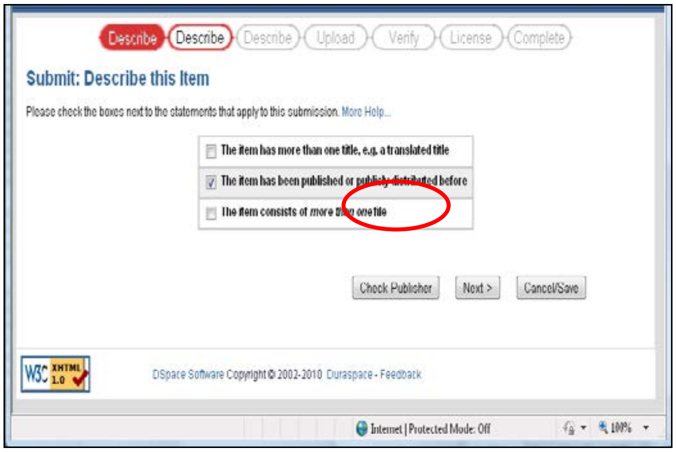
Fig. 1 Diagram of integrating Sherpa Romeo with DSpace

These actions involve taking control from the original application of the DSpace platform and inserting a page in the chain of submitting a new item on the platform. As the DSpace platform is developed by using the JSP (Java Server Pages) technology, namely the HTML pages are generated using functions written in Java language, the sequences newly introduced in the record chain should be preferably written in the same language, using the same JSP technology.

V. SUBMITTING THE REQUEST

The first step in recording a new article consists of the interactive setting of several variables influencing the way in which the following pages are displayed: number of versions of the article title, the number of files to be uploaded and whether the article was published before (in this case, specifying the publisher will be requested).

In this first page, intervention may be made on the JSP code to create a new button, as shown in Fig. 2. The newly-created button is "Check Publisher", circled in red in the figure.



Submit: Describe this Item

Please check the boxes next to the statements that apply to this submission. More Help...

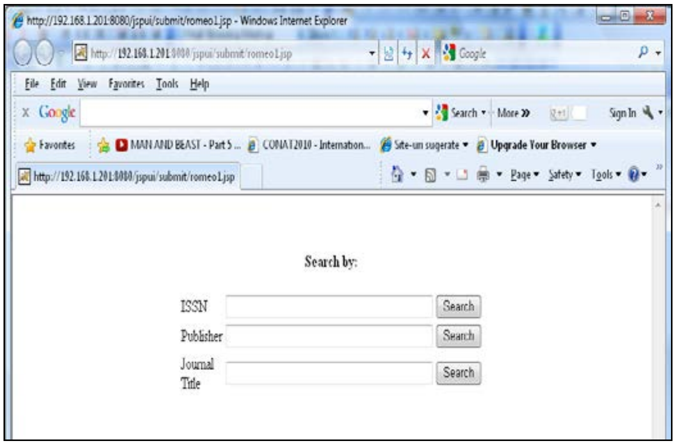
□ The item has more than one title, e.g. a translated title
☑ The item has been published or publicly distributed before
□ The item consists of more than one file

DSpace Software Copyright © 2002-2010 Duraspace - Feedback

Fig. 2 Diagram of integrating Sherpa Romeo with DSpace

Where opening the window is triggered on the onclick action (mouse click on the button) (DHTML), and the transition to the next page is made by the input tag of submit type.

The page opened upon pressing the button will include the query options: ISSN, Publisher, Journal Title.



http://192.168.1.201:8080/jspui/submit/romeo1.jsp - Windows Internet Explorer

Search by:

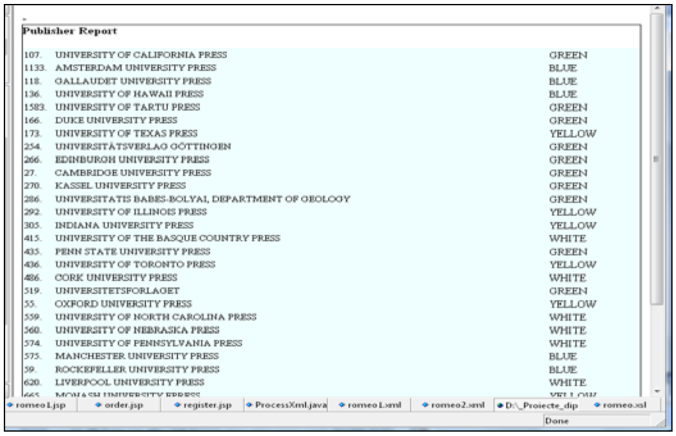
ISSN
Publisher
Journal Title

Fig. 3 Simplified search page

A simple search page by the three criteria is the one presented in Fig. 3. It is basically a form with three input fields and three different submit-type buttons, one for each search criterion. Pressing each button sends the corresponding query to the SHERPA RoMEO server, the parameter being the one specified in the input field concerned.

The transformation results in an HTML file that will look like in Fig. 4. It is a simple list of identified publishers and the "romeo" color associated with each.

With a view to obtaining all the information about the publisher, it should be searched either the exact name of the



Publisher Report

107.	UNIVERSITY OF CALIFORNIA PRESS	GREEN
1133.	AMSTERDAM UNIVERSITY PRESS	BLUE
118.	GALLAUDET UNIVERSITY PRESS	BLUE
136.	UNIVERSITY OF HAWAII PRESS	BLUE
1583.	UNIVERSITY OF TARTU PRESS	GREEN
166.	DUKE UNIVERSITY PRESS	GREEN
173.	UNIVERSITY OF TEXAS PRESS	YELLOW
254.	UNIVERSITÄTSVERLAG GÖTTINGEN	GREEN
266.	EDINBURGH UNIVERSITY PRESS	GREEN
27.	CAMBRIDGE UNIVERSITY PRESS	GREEN
270.	KASSEL UNIVERSITY PRESS	GREEN
286.	UNIVERSITATIS BABES-BOLYAI, DEPARTMENT OF GEOLOGY	GREEN
292.	UNIVERSITY OF ILLINOIS PRESS	YELLOW
305.	INDIANA UNIVERSITY PRESS	YELLOW
415.	UNIVERSITY OF THE BASQUE COUNTRY PRESS	WHITE
435.	PENN STATE UNIVERSITY PRESS	GREEN
436.	UNIVERSITY OF TORONTO PRESS	YELLOW
486.	CORK UNIVERSITY PRESS	WHITE
519.	UNIVERSITETSFORLAGET	GREEN
55.	OXFORD UNIVERSITY PRESS	YELLOW
559.	UNIVERSITY OF NORTH CAROLINA PRESS	WHITE
560.	UNIVERSITY OF NEBRASKA PRESS	WHITE
574.	UNIVERSITY OF PENNSYLVANIA PRESS	WHITE
575.	MANCHESTER UNIVERSITY PRESS	BLUE
59.	ROCKEFELLER UNIVERSITY PRESS	BLUE
620.	LIVERPOOL UNIVERSITY PRESS	WHITE
665.	MONASH UNIVERSITY EPRESS	YELLOW

Fig. 4 Search result by the publisher name

publisher, but it is rather likely to be introduced differently from the Sherpa submission, or the ISSN code. The search by title equally displays a list, as a result, as if searching by the publisher name. In both cases the processing may be extended by the user selecting an item from the list.[25]

VI. Conclusions

The software application has practical implications and represents an original solution to the needs of the academic community. The application is created with a very low cost, both platforms are free to use, being the results of research projects. The proposed model can be also very useful to other universities with the same problems and obstacles in populating and developing institutional digital repositories. A practical implication concerns the easy, one-button, access to two applications simultaneously: archiving, in a digital platform, a published article and accessing the list of publishers enrolled in the platform SHERPA RoMEO's database of publisher policies on open sharing. Another implication is to reduce the time of promoting the digital repository's services and the archiving time. Regarding originality and article value, it is ascertained that a need identified in the self-archiving process is solved, a barrier to the use of digital repository through an original software application.

ACKNOWLEDGEMENT

This paper is supported by the Sectoral Operational Program Human Resources Development (SOP HRD), ID134378 financed from the European Social Fund and by the Romanian Government.

References

[1] J. Allen, "*Interdisciplinary differences in attitudes towards deposit in institutional repositories* (unpublished master's thesis)", Manchester Metropolitan University, UK, 2005.

[2] S. Watson, "Authors' attitudes to, and awareness and use of a university institutional repository". In *Serials*, 20(3), pp. 225-230, November 2007.

[3] L. Chan, "Supporting and enhancing scholarship in the digital age: The role of open-access institutional repositories", in *Canadian Journal of Communication*, 29(3), pp. 277–300, 2004.

[4] E. Gadd, C. Oppenheim and S. Probets, "RoMEO studies 1: The impact of copyright ownership on academic author self-archiving", in *Journal of Documentation*, 59(3), pp. 243–277, 2003.

[5] E. Gadd, C. Oppenheim and S. Probets, "RoMEO studies 2: How academics want to protect their open-access research papers", in *Journal of Information Science*, 29(3), pp. 333–356, 2003.

[6] N. F. Foster and S. Gibbons, "Understanding faculty to improve content recruitment for institutional repositories", in D-Lib Magazine, 11(1), 2005.

[7] N. A. Van House, "Digital libraries and practice of trust: Networked biodiversity information", in *Social Epistemology*, 16(1), 99–114, 2002.

[8] R. Crow, "The case for institutional repositories": A SPARC position paper, in *ARL Bimonthly Report*, 223. Retrieved July 6, 2010, from http://works.bepress.com/cgi/viewcontent.cgi?article=1006&context=ir_research

[9] J. Kim, "Faculty Self-Archiving: Motivations and Barriers", in *Journal of the American Society for Information Science*, 61(9), 1909-1922, 2010.

[10] A. Comsa,I. Maniu,, N. Modler, W. Hufenbach, W., EC Lovasz,, V. Ciupe, An Overview of Library Automation in *Mechanisms, mechanical transmissions and robotics Book Series: Applied Mechanics and Materials,* (162) 583-588 , 2012

[11] E.C.Lovasz, D. Perju, KH Modler, A.E. Lovasz, I. Maniu, C. Gruescu, D.Margineanu,V. Ciupe, ,A. Comsa, Demonstrative Digital Mechanisms Library in Mechanisms, *Mechanical transmissions and robotics Book Series: Applied Mechanics and Materials* (162), 37-46, 2012

[12] V. Ciupe, EC Lovasz, CM Gruescu, High Quality Document Digitization Equipment, in *Mechanisms, mechanical transmissions and robotics Book Series: Applied Mechanics and Materials* (162), 589-596 , 2012

[13] S. Harnad, "Maximizing Research Impact Through Institutional and National Open-Access Self-Archiving Mandates", in *Proceedings of CRIS2006. Current Research Information Systems: Open Access Institutional Repositories*. Bergen, Norway. Jeffrey, K., Eds., 2006 http://eprints.ecs.soton.ac.uk/12093/

[14] A. Sale, "The acquisition of open access research articles", in *First Monday*, Vol. 11 No. 9, October 2006.

[15] F. Singeh,, A. Abrizah and A. Karim, "Malaysian authors' acceptance to self-archive in institutional repositories", in *The Electronic Library*, Vol. 31 No. 2, 2013, pp. 188-207, 2013.

[16] A. Abrizah, "The cautious faculty: their awareness and attitudes towards institutional repositories", in *Malaysian Journal of Library and Information Science*, Vol. 14 No. 2, pp. 17-37, 2009.

[17] K. V. Stanton and C.L. Liew, "Open access theses in institutional repositories: an exploratory study of the perceptions of doctoral students", in *Information Research*, **17**(1) paper 507, 2012. [Available at http://InformationR.net/ir/17-1/paper507.html]

[18] DSpace, Project webpage, 2013 www.dspace.org [accessed on June 3rd, 2013].

[19] C. A. Lynch, Institutional repositories: essential infrastructure for scholarship in the digital age. portal: Libraries and the Academy, 3.2: 327-336, 2003.

[20] M. Ware, "Institutional repositories and scholarly publishing", in *Learned publishing*, 17(2), pp. 115-124, 2004.

[21] P. Dietz, *DSpace 1.7.1 System Documentation. Portal*, www.dspace.org,

[22] ASPECKT DSpace, Transilvania University Institutional Repository Web Page, 2013 http://aspeckt.unitbv.ro, [accessed on June 3rd, 2013]. available at: http://eprints.utas.edu.au/388/1/FirstMondayOct06.pdf

[23] L. Kristick, "Using journal citation reports and SHERPA RoMEO to facilitate conversations on institutional repositories", in *Collection Management*, 34(1), pp. 49-52, 2009.

[24] A. Repanovici, "Intelectual property and open access to information", in *International Conference on intellectual property and information management* (IPM '11), A. Repanovici, and C. Murzea Eds. Transilvania University of Brasov, Romania, 7th – 9th April 2011, pp. 153-158.

[25] Sherpa ROMEO, Project webpage, (2013) http://www.sherpa.ac.uk/romeo/api.html , [accessed on June 3rd, 2013]