Setting up the PORTULAN / CLARIN repository

Luís Gomes **Frederico Apolónia** **Ruben Branco** **João Silva** **António Branco**

NLX-Group, University of Lisbon, Portugal
{luis.gomes,frederico.apolonia,ruben.branco,jsilva,ahb}@di.fc.ul.pt

Abstract

This paper aims at sharing the lessons learned at setting up a CLARIN repository based on the META-SHARE software, which we have just used to develop the PORTULAN / CLARIN centre.

This paper documents the changes and extensions to META-SHARE that were needed to fulfil the CLARIN requirements for becoming a B-type centre.

The main purpose of this paper is to serve as a one-stop guide for teams pondering or having decided to adopt META-SHARE software for setting up their own CLARIN repositories in the future.

1 Introduction

In order to set up the CLARIN repository of the Portuguese network, PORTULAN / CLARIN, following other national networks, we decided to use META-SHARE software to set up our repository software, extending it with the functionalities required by CLARIN for B-centre approval.

These extensions address a number of aspects that we will be reporting on in this paper, namely: metadata harvesting (Section 2), single sign-on (Section 3), persistent identifiers (Section 4) and the user interface (Section 5).

META-SHARE is a network of repositories for sharing and exchanging language data and tools. It is also the name of the software[1] running on these repositories and we will refer to it as MS for short.

2 Metadata Harvesting

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is an application-independent protocol specification[2] for metadata harvesting. The OAI-PMH specification describes the communication between *repositories* and *harvesters*.

A repository must implement this protocol to be approved as a CLARIN B-centre. Furthermore, the metadata harvested from the repository must comply with a profile in the Component MetaData Infrastructure (CMDI) CLARIN registry.

A MS metadata profile already exists in the CLARIN CMDI registry and an implementation of the OAI-PMH protocol was kindly provided to us by the executive staff of CELR at University of Tartu.[3] We made minor tweaks to this implementation in order to make it work with our version of MS, which is slightly more recent (we are using the 3.1.1 branch instead of the 3.0 one). Then, we sent an email to <harvester@clarin.eu> announcing our OAI-PMH endpoint to the Virtual Language Observatory (VLO) and asking to be harvested by the alpha VLO instance.

After the harvesting was concluded we browsed the PORTULAN / CLARIN resource collection in the VLO alpha instance, and we checked the correctness and completeness of the metadata.

[1] https://github.com/metashare/META-SHARE
[2] http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm
[3] https://keeleressursid.ee/en/center/people

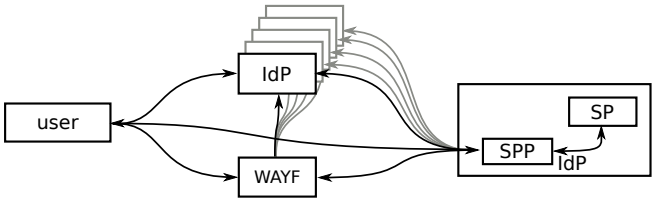

Figure 1: Using a Service Provider Proxy (SPP) to overcome PySAML2 limitations in terms of the number of IdPs it is able to handle.

3 Single Sign-On with SAML2

The Security Assertion Markup Language version 2 (SAML2) is an open standard XML-based framework that enables user authentication within a federation of web Service Providers (SPs) and Identity Providers (IdPs) through Single Sign-On (SSO). In practice, once users have authenticated themselves to a federated IdP, they will thereafter remain authenticated for any service provided by the federated SPs until they explicitly logout.

MS is implemented in Django and djangosaml2[4] integrates the PySAML2 library as a Django application and authentication backend, making it easy to integrate MS as a SAML2 SP. Unfortunately, the PySAML2 library is unable to cope with a large number of IdPs, and when we tried to load the metadata file of the CLARIN federation IdPs a "too many open files" exception was raised (apparently PySAML2 opens many auxiliary files to process a single large metadata XML). Thus, to work around this PySAML2 limitation, we set up a Service Provider Proxy (SPP) using SimpleSAMLphp[5].

As depicted in Figure 1, the proxy behaves as an IdP to our SP (the only IdP that the SP must known about) and behaves as a SP to the IdPs in the CLARIN federation. The local copy of the federation IdPs metadata XML is synced to the master metadata file[6] on a daily basis.

When a user tries to access a service provided by a federated SP, the SP needs to contact the IdP of the user's home institution/organization to authenticate the user and obtain identity attributes such as the user's name and email address.

The number of IdPs in a federation such as CLARIN may be over one thousand and thus a special service called *WAYF* or *Discovery Service* is needed to allow the users to *easily* select the IdP of their home institution from such a large list. Instead of simply presenting a list containing all IdPs in the federation, a good WAYF service will try to automatically detect the users' institution based on several parameters such as the IP address and geo-location of the computer. Also, the user's choice of IdP will be memorized by the WAYF service and recalled upon subsequent visits.

Since we have deployed SimpleSAMLphp as a SPP, we could also use its WAYF implementation. However, we have configured our SPP to redirect users to the CLARIN WAYF service[7] instead, which is based on DiscoJuice[8] and we find it to provide a more user friendly interface than the WAYF of SimpleSAMLphp.

4 PIDs

The use of Persistent Identifiers (PIDs) for resources is a requirement for B-centre repositories.

Fortunately, MS has a metadata field for storing externally-assigned identifiers such as the PID of a resource. Thus far, we have not automated the generation of PIDs for resources deposited in the repository, but we have plans for integrating MS with ePIC[9] through its RESTful API, allowing easy generation of a PID from the metadata editor interface.

[4] `https://pypi.org/project/djangosaml2/`
[5] `https://simplesamlphp.org/`
[6] `https://infra.clarin.eu/aai/prod_md_about_spf_idps.xml`
[7] `https://www.clarin.eu/content/clarin-central-discovery-service`
[8] `http://discojuice.org/`
[9] `http://www.pidconsortium.eu/`

5 Changes to the User Interface

MS project originally used the Blueprint CSS framework. However, since this framework was last updated on 14th of May 2011 and is missing a lot of modern website design features such as responsive design, we decided to change the MS interface to a more modern and robust alternative. As CLARIN has made available[10] an implementation of the interface guidelines as a Bootstrap theme, we decided to use this theme as the basis of our work on revitalizing the MS interface.

Bootstrap is easy to use, well documented and supports responsive design. This has made it very popular and is currently supported by a large community. Because the aforementioned CLARIN theme was built upon Bootstrap version 3, we are using that version instead of the newest version 4.

The CLARIN human interface guidelines[11] specify, among other things, the general page layout, the typography that should be used and the colour palette to be followed. With respect to the layout, MS already provides a very solid foundation but some changes are required to make PORTULAN / CLARIN comply with the guidelines.

The landing page of the repository allows searching and browsing the resources by means of a search box and a list of "Filter by" tags which implement a faceted search. Following the CLARIN guidelines that suggest secondary navigation to appear on the right-hand side of the page, these "Filter by" tags were moved to the right, as shown in Figure 2 (a).

The search box was sized down and moved to the upper right corner of the page (b).

In MS, the user had to hover the entry title to see a description of the resource, but we opted to make the description always visible, although clipped to a maximum of 300 characters, requiring less effort from the user to get to important information (c).

Furthermore, different resource and media types were conveyed to the user by adorning resource titles with different icons, which requires the user to hold a memory mapping between icons and their meaning. We replaced these icons with their textual meaning in a key-value structured presentation, which we find easier to assimilate, as shown in (d), (e) and (f) in the figure.

To minimize scrolling on today's wide screen monitors, we resized the page header to take up less vertical space, as shown in (g).

6 Conclusion

The repository of the PORTULAN / CLARIN centre is ready to enter production. We keep doing tests and rounding some sharp edges, but the main requisites for a B-centre repository seem to be ensured.

Ideally, MS should be ported to make use of more recent and supported versions of its dependencies, in particular Django, but the effort required is too high at this point. Therefore, to mitigate possible security flaws in MS and its outdated dependencies we decided to keep our repository running within a virtualized environment accessed through a reverse HTTP proxy. Despite the disadvantage of not being actively developed and supported, MS has the advantage (for us) of being implemented in Python and Django, which allows us to reuse knowledge of this language and framework in other components of the CLARIN / PORTULAN centre.

The next main step will be the CoreTrustSeal quality assessment procedure.

Acknowledgements

We thank the executive staff of CELR at University of Tartu[12], especially Neeme Kahusk, for providing us with their MS implementation of the OAI-PMH protocol and SAML2 configuration. We thank Esmeralda Pires from the FCCN (Portuguese Foundation for National Scientific Computation) for helping us with SAML2 questions.

[10] https://github.com/clarin-eric/base_style
[11] https://office.clarin.eu/v/CE-2016-0794-CLARINPLUS-D3_1.pdf
[12] https://keeleressursid.ee/en/center/people

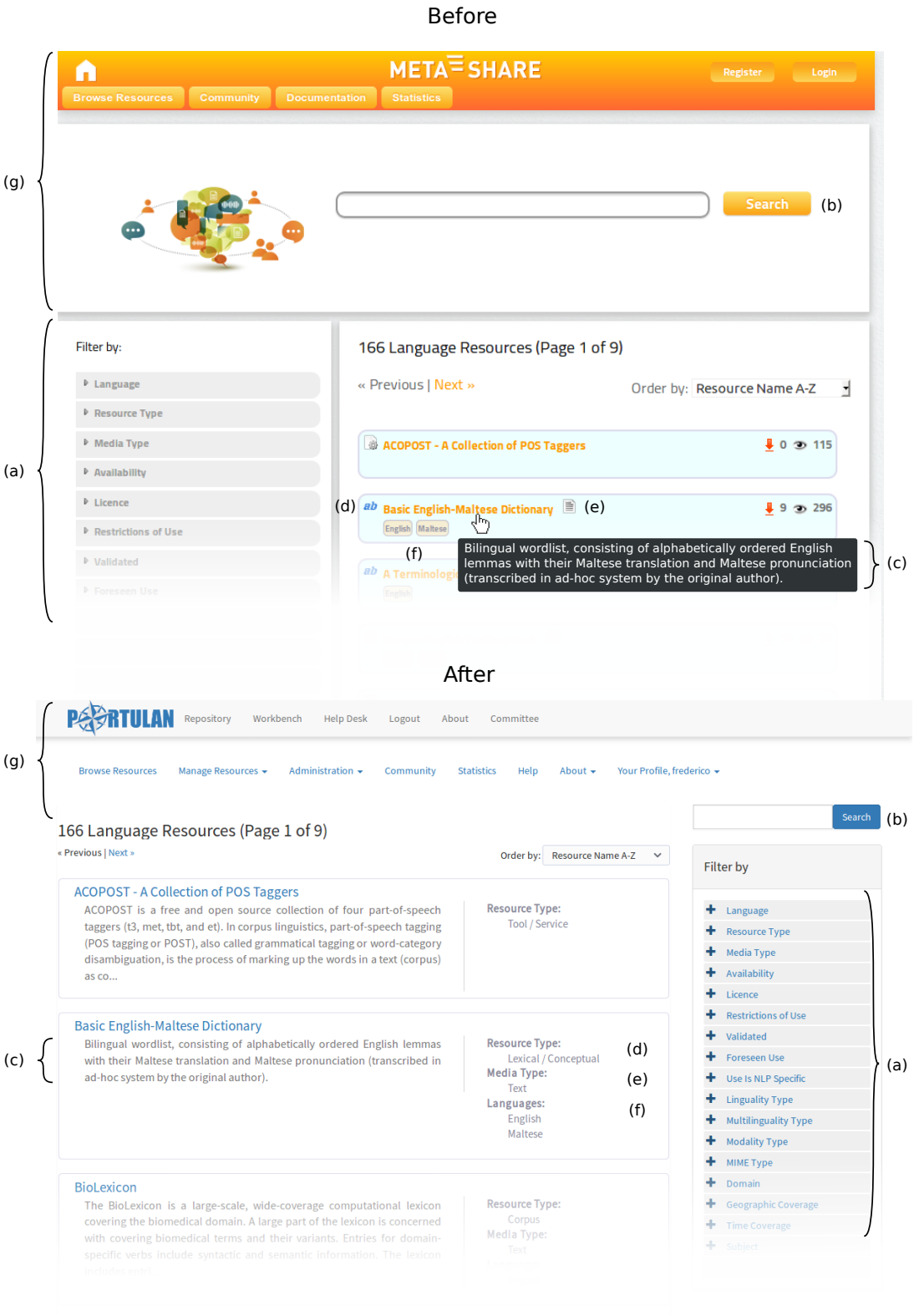

Figure 2: Repository search/browse page before and after our changes.