Service Cloud for information retrieval from multiple origins

Authors:

Marisa R. De Giusti, CICPBA (Comisión de Investigaciones Científicas de la provincia de Buenos Aires), PrEBi, National University of La Plata, marisa.degiusti@sedici.unlp.edu.ar
Ariel J. Lira, PrEBi, National University of La Plata, alira@sedici.unlp.edu.ar
Nestor F. Oviedo, PrEBi, National University of La Plata, nestor@sedici.unlp.edu.ar

Abstract *— One of the main intents of all digital libraries is to provide wide access to academic resources. Several tools and techniques are available to achieve this goal, some of them focus on harvest and storage of resources while others on retrieval and presentation. The heterogeneity of protocols, metadata formats and storage mechanisms generates a set of isolated datasources, which can be separately accessed by custom protocols and which usually contain only one part of the available information. This project suggests a solution for digital libraries based in a hybrid combination of software tools. The solution makes the procedure easier of information retrieval from heterogeneous datasources (OAI-PMH data provider, web services, z39.50, local repository, etc.) that can be queried from an arbitrary number of interfaces (OPAC, web services, z39.50, etc). Each of these entry points represent an unified view of the whole, which allows to easily interoperate across the interfaces and to collect, organize and return the results to the user that made the request. The development of this solution is based on the configuration of a set of special purpose tools. Additionally a set of adapters or drivers have been created, to allow the interoperability from and to each of the mentioned tools. The following tools have been considered so far in this project: an OAI harvester for harvesting bibliographic resources in any metadata format, Solr servers to index and retrieve information efficiently, a public OPAC, and finally a federated metasearch engine which works over the SRU/SRW and z39.50 standards. It is also possible to include other tools, which enables the opportunity add new services and data sources, without redesigning existing interfaces. Concepts developed along this work have been put into practice inside the Intellectual Creation Dissemination Service of La Plata National University. This initiative copes with a wide range of services for professors, researchers and students, such as: assisted search and provision of academic material in ISTEC and some non-ISTEC libraries, management and exposure of the institutional repository through OAI-PMH, public website to expose local and external resources (obtained via OAI-PMH) and harvesting of resources housed in academic units' libraries of the University.*

Index Terms *— heterogeneous information, information retrieval, multiple data sources, service integration*

INTRODUCTION

To offer a greater number of academic resources to their users, Institutional Repositories usually make use of other repositories' resources as search source. These resources can be fully imported into the local repository or can be online accessed through web-services. A hybrid approach can be used too, in which some repositories are fully imported while others are searched remotely. In any case, the way the resources are fetched and gathered must be transparent for the users.

The interaction among different repositories faces the developers with some problems, many of them with no standard solution yet. Different communication protocols, heterogeneous data formats, specific and restricted storage methods, are just a few of them. Services and resources should be integrated to solve these problems. Nevertheless, the solution must ensure that the addition of new services will not require to modify the whole system.

The Intellectual Creation Dissemination Service (Servicio de Difusión de la Creación Intelectual, SeDiCI) [1] is an initiative of National University of La Plata (UNLP) responsible for gathering and publishing academic and scientific resources (such as thesis, papers and proceedings) produced inside the University. SeDiCI also collects and exposes resources retrieved from other institutional repositories from all around the world. In order to efficiently carry out these activities, it has been defined a process for fetching those resources with a unified storage mechanism to provide an agile search and retrieval service.

Besides SeDiCI, UNLP launched in 1997 the Library Linkage Project (Proyecto de Enlace de Bibliotecas, PrEBi) [2]. PrEBi offers a bibliographic search service to users (teachers, researchers and students) of UNLP. This service is supported by a user request management software application called Celsius [3] as well as an integrated search tool which allows performs simultaneous searches over online catalogs. This tool includes among its catalogs all external resources fetched by SeDiCI.

PrEBi and SeDiCI projects have been deployed to help UNLP members in their everyday work. The former plays the role of resource provider (looking for specific bibliographic resources), and the later publishes high quality academic resources on an online website.

This paper describes the different tools and methods used for storing, indexing, retrieving and sharing bibliographic resources. It explains how these tools are used for services provided by PrEBi and SeDiCI. Furthermore, it presents some improvements that contribute to the quality of cataloging work and services.

MAIN GOALS

UNLP users are characterized by their need for very specialized academic resources, inside the scope of their research areas. In order to satisfy those needs, a set of primary objectives has been maiden to cover it as far as it is possible. These objectives are:

- to build a simple tool for harvesting Open Access [4] resources (via OAI-PMH [5]), which automatizes the harvesting process;
- to have a federated search engine capable of performing simultaneous searches over many library catalogs (that support at least the Z39.50 [6] and SRU/SRW [7] protocols) and Open Access resources gathered by the harvesting processes;
- to define efficient methods for indexing and retrieving documents, retrieved either via OAI-PMH or from SeDiCI;
- to have a simple search tool for all the documents, which permits the addition of semantic like search functions to obtain truly relevant results;
- to provide a mechanism for retrieving resources from the UNLP schools, and importing them into SeDiCI. The actual idea is to make simpler all cataloging tasks and to provide extra flexibility to review and correct the resources in order to improve their quality.

Building independent solutions for these objectives may be, at first glance, a fast and easy idea. However, it will require the replication of data and the repetition of some tasks, leading as a consequence to less scalable and harder to maintain solutions. In contrast to that model, a set of tools with very specific functions is proposed here. These tools collaborate one another to meet the objectives.

There is an OAI Harvester among these tools that fetches resources through the OAI-PMH protocol and applies some transformations to them before storing them into an indexing engine. There is also an OPAC (Online Public Access Catalog) to search over that engine via the REST protocol [7]. A federated search engine has been set up too, which also uses the indexing engine -just like another catalog- through an adapter application that transforms the SRU requests to REST requests.

Most UNLP schools expose their resources to the local network through OAI-PMH. For that reason the OAI Harvester can also be used to fetch and import these resources into SeDiCI. Given that SeDiCI resources are also exposed via OAI-PMH, they can be collected by the harvester tool and indexed by the engine which enables to make the most of the search capabilities.

This tool integration constitutes a Service Cloud which provides an economic and efficient solution to the problems derived from handling heterogeneous resource sources [8]-[9]. In the following paragraphs these tools are described and analyzed.

TOOLS

Each software tool considered in this work deals with a specific part of the problem. All but the OAI harvester are free and open source applications. The harvester was explicitly built by PrEBi developers to complete the so-called Service Cloud. Table 1 shows these tools included into a service type.

Service Type	Software Tool
Federated Search Engine	Pazpar2 Masterkey
Indexing Engine	Apache Solr SolrSRUServer
OAI Harvester	Celsius Harvester
Search Interface	OPAC

TABLE 1
SERVICE TYPES AND SOFTWARE TOOLS PARTICIPATING IN THE SERVICE CLOUD

Federated Search Engine

The main objective of the federated search engine is to offer a simple search interface for accessing several online catalogs simultaneously. Results are joined and displayed to the user altogether. This tool was built from two Open Source applications: Pazpar2 and Masterkey, as back end and front end respectively. Both applications were created by Indexdata.

Pazpar2 is responsible for metasearching over a set of pre-configured catalogs, and supports the most common protocols for online search and retrieval of bibliographic resources: Z39.50 and SRU/SRW. For each search request, Pazpar2 simultaneously connects to each catalog to send the request. When responses start to arrive, it applies a XSL transformation to each one to map the incoming data into a generic normalized format. Afterwards the normalized results are stored in a temporal memory. Once all results have arrived, they are merged and duplicated records are deleted from the temporal set. The records set is then sorted according to some predefined criteria (the relevance is the default). In the meanwhile, Masterkey presents to the user partial results as they arrive, which allows the user to preview and to access any resource with no need for the search process to be finished.

Pazpar2 also provides the *faceting* function. *Facets* are content summaries for some specific fields of the metadata format. These summaries are displayed for every *facetable* field, showing the *N* most frequent terms in that field. This option is very useful to refine the search using the original result set as criteria. For example, a refinement on the field *Subject* will return only the results matching the selected Subject.

Pazpar2 configuration flexibility results useful to specify many configuration parameters, such as:

- a generic metadata format to which all results will be transformed;
- the criteria used to detect duplicated records;
- the criteria to calculate the records' relevance;
- configuration specific for each catalog: search fields, metadata formats, encoding, etc.

Since Pazpar2 is just a back end accessible through web services, it can be used from any web services-enabled front end application. Masterkey was chosen as front end, given that it is Open Source too and it was specially designed to work with Pazpar2. Masterkey has a customizable interface based on XML and CSS. Its internationalization capabilities are simple and powerful as well. This front end includes all the elements needed for a metasearch application: simple and advanced search, selection of catalog, faceting, pagination and sorting. It also has an administration section to manage the catalogs, to define catalog collections and to specify access restrictions either by username or IP range.

Indexing engine

When dealing with large information sets, it is necessary to index it in order to improve search and retrieval process. For this project it has been selected an indexing engine called Apache Solr [10]. Solr is a java-based server developed by the Apache Software Foundation. Its robustness and high performance have been extremely useful for many organizations to search among millions of records in amazing response times. Another advantage of this engine is the use of a very common communication protocol like REST.

Apache Solr uses a *schema* configuration to define the field structure to represent the documents in the index. This schema can also be used to specify the process that has to be applied to the content of each field before its indexing. The

schema configured for this Service Cloud is analogue to the generic metadata format used in the metasearcher. This analogy makes it simpler the interaction between these two applications.

Apache Solr has another important feature called *"cores"*. Each Solr *core* is an independent index and has its own configuration. There only exist one Solr server instance that listens for requests and maps them to the different cores running under that instance. This is useful for splitting the data into different sets and for optimizing available computing resources. The use of Solr cores will be shown in the OAI Harvester Section.

Scalability is a request for this kind of applications. Fortunately Apache Solr can work in a distributed environment which makes it highly scalable. It also permits to install and configure plug-ins, adding new and improved search methods. For example, with a semantic like plug-in and a suitable configuration, it is possible to perform semantic like queries with a relatively low cost.

Clearly Apache Solr is one of the most important components of the Service Cloud. It is the main storage for almost all information fetched from any repository and it provides many great functions to improve both in performance and efficiency the information retrieval tasks.

Apache Solr as a catalog inPazpar2

As mentioned before, harvested resources are also used in the metasearches. To make this possible, it was used an adapter tool called SolrSRUServer which can receive SRU requests and translate them into REST requests. This way, Pazpar2 can search via the Apache Solr server as if it was just another SRU catalog.

Given that there might exist multiple Apache Solr servers running, it was necessary to modify the SolrSRUServer a little bit to avoid having one instance for each Solr server. Now, the adapter receives in the requests an extra parameter that contains the URL of the Solr server that will attend the request. This additional parameter permits to have one single instance of adapter mapping requests to different SorlSRUServer servers.

OAI Harvester

Basically, an OAI Harvester is a software application that interacts with OAI Data Providers [5] via the OAI-PMH protocol and retrieves from them a set of Open Access documents. For this Service Cloud it has been used Celsius Harvester, an OAI harvester developed inside PrEBi. This application is currently working and has already downloaded more than 12 millions of documents.

Developed with the purpose of providing a simple and intuitive management tool, even for users without technical knowledge. It aims to manage all the concerns related to harvesting, including many interesting features such as:

- **Scheduled Harvesting:** a scheduled harvesting is a sequence of individual harvesting tasks. The execution of these tasks allows to harvest all exposed resources in the repositories without user intervention. The local copy of the repository is updated automatically every certain period of time;
- **Normalization:** Although the OAI specification forces Data Providers to expose their resources at least in Dublin Core, but does not defines the way it must be used. Because of this, Celsius Harvester allows to upload a transformation file (XSL) for every repository. Thus, every downloaded document from that will be normalized. If no file is uploaded, the documents will be processed as they are retrieved;
- **OutputHandler:** they are responsible for processing and deciding the final destination of the retrieved documents. So far there exists one OutputHandler that transforms and inserts records into Solr and other one that splits a multiple-document file in many single-document files. The OutputHandlers allow to add new functionality to Celsius Harvester with no need for new applications;
- **Collections:** collections are a simple way to group repositories under some criteria. Each group has an OutputHandler configured that processes all documents. The following collections have been defined:
    - **External:** documents retrieved from Open Access repositories external to the UNLP. This collection uses an OutputHandler to index the documents in a specific Solr core.
    - **Internal:** documents harvested from the UNLP schools. The OutputHandler for this collection splits server responses to create individual files for each document.
    - **Sandbox:** documents harvested from any repository which require a revision to ensure or improve their quality. It uses the OutputHandler to index documents into a specific Solr core.
    - **SeDiCI:** harvesting tasks are made from SeDiCI and gathered documents are indexed in a specific Solr core. This permits to enhance search services provided by SeDiCI.

OPAC

The proposed OPAC takes into account many functionalities to provide a complete tool for searching and exploration. It includes contextual information for helping users to search without lost of simplicity. Even though its main purpose is to offer a searching front end, it is being considered its inclusion inside other applications.

According to the objectives previously presented, this OPAC -still in a development stage- must have at least the following functionality:

- **related documents:** similar keywords, similar titles, same authors, subjects closely related, etc;
- **related authors:** other authors that have written about the same subject, co-authors in other documents, etc;
- **related searches:** documents from other searches with similar queries;
- **search history:** documents accessed by other users with similar queries;
- **tag cloud:** cloud of words with most frequent terms;
- **search suggest:** correction of terms, similar words, similar phrases, old searches, etc.

Most of these features can be implemented using Apache Solr capabilities, which ensures efficiency and simplicity to the OPAC.

Internal resources imported into SeDiCI

As mentioned above, UNLP has 17 schools; most of them have their own libraries, which are managed by themselves. Thus it is necessary an easy way to speed up while keeping simple the mechanism for incorporating resources from every Academic Unit. The implemented solution consist of a semi-automatic harvesting mechanism based on the OAI-PMH protocol. This service is supported with Celsius Harvester tool. UNLP Schools just need to implement a relatively simple OAI Data Provider, and use it to expose their resources. With this simple interface, Celsius Harvester can now harvest and index all exposed resources. A simple authentication method such as HTTP Digest [11] can be included as well, adding additional security to this implementation. All retrieved documents are imported into SeDiCI, starting in revision and correction stages, to be then included into SeDiCI public resources.

GENERAL OVERVIEW

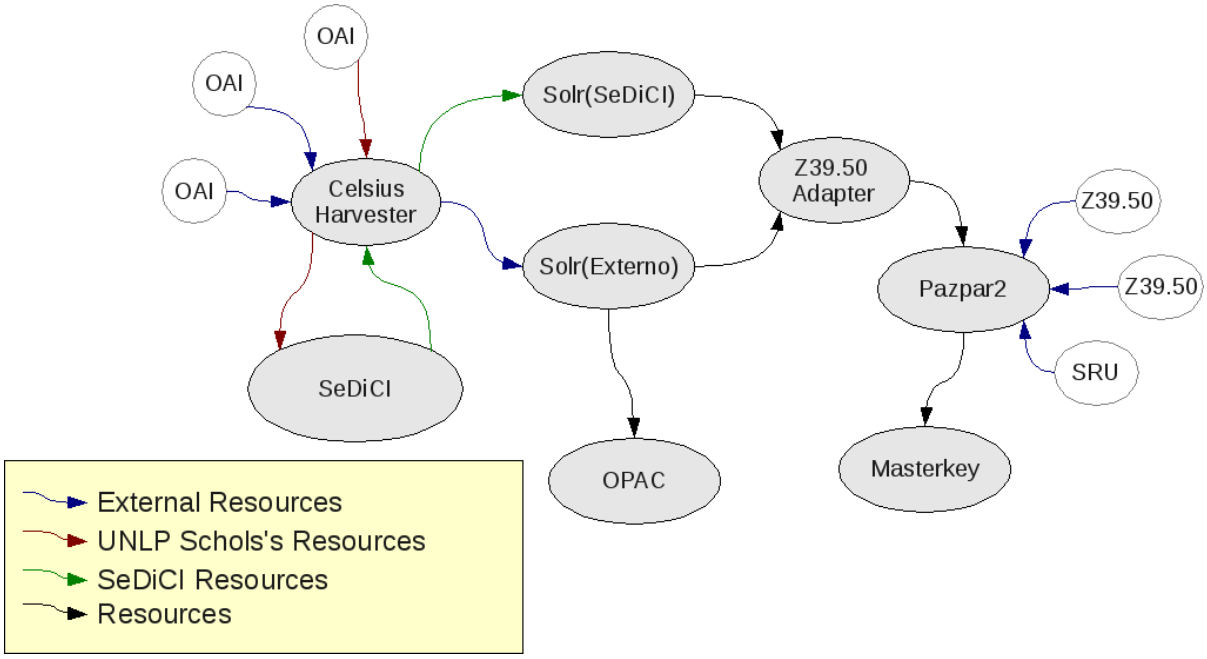

FIGURE 1
SERVICE CLOUD SCHEMATIC VIEW

Conclusion

The project presented in this work proposes many stages to make the best of all available digital resources for UNLP researchers, students and teachers: to automatize resources retrieval from OAI Data Providers via the Celsius Harvester tool, which includes UNLP and even SeDiCI resources; to normalize all fetched resources attempting to avoid most of problems derived from data heterogeneity; and to distribute the data among different storage supports.

It is also remarkable the use of an indexing engine such as Apache Solr server, and some of its features (such as the inclusion of *cores*). A wide range of functions in the information retrieval field can be easily added with these features, allowing other tools such as an OPAC to make use of many of them. Harvested documents can also be used by the meta-search tool Pazpar2 via the SolrSRUServer connector.

Solutions presented here as a Service Cloud use only free and Open Source software. Besides they are quite simple to implement, requiring some technical skills and hardware present in any modern computer (Apache Solr server might require a more powerful computer). This makes this cloud a low-cost, flexible and scalable alternative for digital libraries that need to maintain and add services for users, different data sources or just local digital resources. The use of different applications that work together allow to use all or just few of them, which can be adapted to cope with the different situations of each repository.

Refrences

[1] Servicio de Difusión de la Creación Intelectual, *http://www.sedici.unlp.edu.ar*

[2] Proyecto de Enlace de Bibliotecas, *http://www.prebi.unlp.edu.ar*

[3] De Giusti, M, R, Lira, A, J, Sobrado, A, Inafuku, F, "Celsius Network", *ICEE 2007 Coimbra Portugal*, 2007.

[4] Coleman, A, Roback, J, "Open Access Federation for Library and Information Science", *D-Lib Magazine*, Vol 11 No 12, 2005.

[5] Barrueco, J, M, Subirats Coll, I, "Open archives initiative. Protocol for metadata harvesting (OAI-PMH): descripción, funciones y aplicaciones de un protocolo ", *El profesional de la información*, Vol 12 No 2, 2003.

[6] Moreno, A, Carrión Gútiez, A, Matínez Gallo, J, C, "El z39,50", *CLIP Boletín de la SEDIC*, Vol 30, 1999.

[7] Hickey, T, B, "Web Services for Digital Libraries", *ELAG 2003 - 27th Library Systems Seminar - Bern (Switzerland)*, 2003.

[8] Goh, C, H, "Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Systems", Phd MIT, 1997.

[9] Buccella, A, Cechich, A, Brisaboa, N, "An Ontology Approach to Data Integration", *Journal Computer Science & Technology*, Vol 3 No 2, 2003.

[10] Apache Solr, *http://lucene.apache.org/solr.*

[11] Morgan T, D, "HTTP Digest Integrity", *VSR Publications*, 2008.