Towards an Abstraction Layer for Scientific Services

Marcos Baez

Supervised by Fabio Casati

University of Trento – Dipartimento di Ingegneria e Scienza dell'Informazione
Via Sommarive 14, 38100 Trento, Italy
baez@disi.unitn.it

Abstract. This paper presents a platform that facilitates building scientific services on top. We describe the problems in building such services and derive a general-purpose, extensible layer for accessing any resource that has a URI and is accessible on the Web. The platform, and more in general the classes of systems that have this functionality, is referred to as Resource Space Management System and is the (scientific) resource analogous of a data space management system. In this paper we describe the model and conceptual architecture of the platform, discuss its benefits and outline the research plan for its realization.

Keywords: Resource Space Management Systems, Scientific Resources, Scientific Services

1 Introduction

With the advent of the web era we have moved away from printed papers and journals towards digital formats, as a result, a large number of services allowing their dissemination, archival, sharing and reviewing have emerged. This has also made possible the rising of other non-conventional types of *scientific contributions* e.g., video, datasets, and other resources; very rare before the web. Thus, the Web has opened a world of possibilities for how scientific knowledge dissemination, creation and evaluation could be done and for how the notion of scientific contribution could evolve to serve the need of scientists to learn about novel, interesting research ideas and results.

There has been a considerable amount of work, ranging from theoretical to practical proposals, on how to exploit the Web to improve the way we do science today. Perhaps, one of the most representative of such initiatives, due to is ambitious goals and scope, is *Liquidpub*[1]: an EU project that aims at capturing the lessons learned and opportunities provided by the Web and open source, agile software development to develop concepts, models, metrics, and tools for an efficient (for people), effective (for science), and sustainable (for publishers and the community) way of creating, disseminating, evaluating, and consuming scientific knowledge [1].

Besides concepts and models, from a technological/service perspective, implementing the vision of initiatives such as Liquidpub require the development of *servic-*

[1] http://project.liquidpub.org

es for, among others, i) supporting scientists in knowledge search, aggregation, and evaluation by interfacing with "traditional" or "novel" data sources (from SpringerLink and ACM repositories for access to scientific data and metadata to social bookmarking sites such as citeUlike), ii) with systems that support the knowledge evaluation process (e.g., conference management systems), and iii) with systems that provides for early sharing of knowledge (blog, wikis). These are just some examples of the arbitrary number of scientific services that can be built once access to scientific resources is available.

Given the above, we need an extensible and common platform to access the various kinds of scientific resources available on the web, that makes it easy (or at least easier) to develop services on top. The goal of this research work is to design and develop such a platform; which we refer to as a *resource space management system* (RSMS). We plan to achieve this by providing the RSMS with the following characteristics:

- **Homogenous programmatic access to scientific resources and web services** regardless of how they are implemented as long as they are web accessible (via browser or rest/soap API).
- **Universality**. We aim at covering a large set of scientific resources of various kinds as described above. While anything identified by a URI is in scope of RSMS (whether it is a scientific resource or not) we aim at provide concepts and services for scientific resources, such as built-in notions of authors, references, and the like.
- **Collaborative Extensibility**. Given the large amount of services available, it is practically impossible to provide a monolithic infrastructure that incorporates all of them. We made an early design decision to facilitate extensibility by the community where developers can just register services that interface with systems such as scientific resources and that may be hosted within RSMS but also by other parties (i.e., there is no need for plugging code in).

Building such an infrastructure presents several interesting issues and challenges not only from a practical but from a conceptual point of view. These issues and their implications (that will be addressed on this work) can be summarized as follows:

- **Heterogeneous interfaces.** Scientific resources can be provided by "traditional" or "novel" data sources. Interfacing with such heterogeneous data sources normally requires clients to implement the access for each of them, given that the interfaces of these sources have different signature details. In some cases, sources are not even exposed via APIs or meant to be crawled (for example, getting citations from Google Scholar), then requiring clients to implement wrappers and raising some other problems like limited access, banning etc. Therefore, to facilitate building services on top, the platform should abstracts the specifics of the different data sources and provide a common interface.
- **Lack of common conceptual model for scientific resources.** Given the vast number of resources in the "space" of resources, it becomes difficult for services to handle each resource-specific operations (e.g., searching, publishing, changing access rights) and properties (metadata). The lack of a common conceptual model for resources makes services to be limited to specific resources, i.e., the ones hard-coded in the service implementation. It also means that clients need to integrate the different data sources and give the semantics to the resources and their relations.

What this implies is that a common conceptual model should be general enough to cope with the potential requirements of the services on top, and simple enough to be useful. Defining such a conceptual model with a proper compromise is part of this research work.

- **Difficulty to extend sources available.** Extending the sources available to the services implies in most cases changing the service implementation to introduce the required support (e.g., adding a new citations source to a service that computes citation-based indexes). To avoid this problem, the platform should incorporate an extensibility model that allows extending the sources available without introducing changes in the platform.
- **Maintenance cost and scalability.** Associated to the above problems is the maintenance cost. Adding new sources, maintaining wrappers as they may become obsolete, providing support to new resources; all of them imply effort. Moreover, as the number of services grows, the scalability of the platform becomes a problem. Therefore, to sustain a platform like the one we intend, it is necessary to reduce the effects on costs and performance. A preliminary model based on distribution of effort and computation is presented on this paper.

In the following we outline a research plan towards the design and development of models and systems for RSMSs.

2 Use case: Liquid Journals

As part of this research work, it will be developed a use case based on a new model for scientific knowledge dissemination: *liquid journals* (LJ) [2]. This use case will allow us validate the concepts, models and system to be developed.

In a nutshell, LJ is a new dissemination model capable of bringing "interesting" and "relevant" scientific contributions that can be found on the web. In this sense, we support the idea of *searching* over *submission*, that is, the "interesting" content is retrieved typically by querying the Web for scientific contributions (this includes querying traditional, peer-reviewed journals).

Thus, the effort in developing the liquid journals will be on the definition of a query language capable of capturing the notions of "interestingness" and "relevance", and on the development of the underlying query engine on top of the RSMS, capable of merging results from various data sources (e.g. search engines, social bookmarking services, …), filtering and grouping the results according to the query definition and to rank them according to their relevance. The RSMS will provide seamless access to the scientific data sources and a conceptual model for scientific contributions.

3 Research Space Management Systems

In order to overcome the issues we have presented before, we build the platform around the abstraction of scientific resource and provide a general and extensible model: the *resource space*. Then, from the infrastructure point of view, we provide a

collaborative-extensible and distributed platform. An overview of the preliminary model and platform is presented in the following.

3.1 The Resource Space

RSMS is based on the notion of viewing every possible kind of scientific contribution available on the web as a scientific resource. Under this assumption, the web is a (scientific) resource space and the RSMS manages – and simplifies – access to these resources.

A *resource* can be any artifact we can refer to by an URI and that is accessible over the Web (e.g., documents, experiments, but also metadata from citeUlike and Google Scholar, etc). These resources are managed by potentially different service providers (e.g., Google Docs, Google Scholar, ...). We refer to these service providers as *resource managers*. Then, the third element we consider is the *action*. Actions describe the services provided by resource managers and that allow us to operate with the resources (e.g., to share or search documents, or more complex actions such as crawling a web site for scientific metadata). On top of this we provide set of abstractions, to free upper layers of implementing resource specific operations.

Incidentally, these abstractions are natural extensions of the basic elements. Thus, the first abstraction we consider is the *resource type*, which characterizes families of resources with similar behavior. Analogously, *resource manager types* denote general classifications of resource managers, such as archives, search engines, control version systems, etc. Then, the *action type* provides a common interface for semantically equivalent actions. For example, to "change access rights" in both Wiki and GoogleDocs regardless the differences in their "signature" detail.

On top of these constructs, we define entities and specific metadata and operations for scientific resources that correspond to common resources and actions that services need to perform. In this preliminary model, we consider the following scientific entities: scientific contributions, people, communities, events. The mapping of those entities to the resource space is performed by defining particular resource types that encapsulate the properties, relations and behavior of those. It is possible, however, to extend these entities following our extensibility model below.

3.2 Extensibility model

In general terms, the approach we follow is to provide a set of core modules that can manage the *adapters* and access to resource managers through these adapters. Adapters are provided by third parties and made available to the upper layers trough the registration service of the RSMS. This allows us to extend the sources available without introducing changes into the platform, so making the platform easier to maintain.

The RSMS extensibility approach, the resource manager and the concept of resource type collectively support a flexible binding approach that can range from static to dynamic binding to both adapters and (for services using the RSMS) to resources. Static binding to adapters is implemented by restricting (for a given or all RSMS

clients) access to a given (set of) resources to go through a specified adapter - and therefore using a specific mapping between generic actions and actual operations.

However in general it is possible to change dynamically the adapter we use to access a given resource: the mappings are specified and the adapters are registered, this is transparent to RSMS clients. Besides load balancing, the key benefit here is reliability and the ability to leverage the community to maintain a complex distributed system. For example, the RSMS could switch to another adapter in case the one in use becomes obsolete. Note that dynamic binding here is "provider-enabled" in that the provider of the adapter makes sure to define the mapping with the resource type actions.

4 State of the Art

With the goal of providing access to scientific resources available on the web, search engine technology has been explored and applied to scientific content [3]. Specialized search engines have been developed for searching papers /books across multiple repositories using crawling techniques and protocols. Google Scholar[2] and Citeseer[3] are classical examples of scientific search engines that use crawling as technique. BASE[4], on the other hand, is an example of an academic search engine that indexes the metadata providing from repositories which implement the OAI-PMH[5] protocol. This protocol, however, is limited to dissemination of content (metadata from repositories), which is only one angle of the problems we are facing in this work. Another proposal in [4], proposes a framework to support distributed digital information service such as digital libraries. However, this proposal does not address the problem of accessing scientific resources disseminated on over the internet, and the problem of extending the type of services available to upper layers.

It is worth mentioning services such CiteUlike[6], SciLink[7], SciSpace[8], Mendeley[9], Zotero[10], which provides scientists with the tools for organizing, sharing papers, creating social communities and making contacts. These services, although very interesting and useful, do not provide a reusable infrastructure to build other applications on top. They contribute however with scientific resources that could be used by scientific services.

The most relevant work to RSMS is that of *Dataspaces*, which extend DBMS concepts to reach heterogeneous data sources [5][6]. In particular, building applications over this layer allows searching and operating with multiple data sources using a common interface. As this framework is general, therefore, it does not provide any

[2] http://scholar.google.com/
[3] http://citeseerx.ist.psu.edu/
[4] http://base.ub.uni-bielefeld.de/
[5] Open Archives Initiative Protocol for Metadata Harvesting
[6] http://www.citeulike.org/
[7] http://www.scilink.com/
[8] http://scispace.net/
[9] http://www.mendeley.com/
[10] http://www.zotero.org/

particular modeling for scientific resources. In addition, there is not focus on extensibility, on providing resource type abstractions, which hides the specificities of the resources and allows operating on resource of the same type with the same set of operations.

5 Conclusion and Research Plan

In this paper we have introduced a conceptual framework for RSMS. This system is inspired on the idea of having a homogeneous view of a space of resources, in which those resources can be provided by different and heterogeneous resource managers on the Internet. In this context, we have also introduced a preliminary model for the resource space applied to scientific entities. The innovative aspects of the proposed abstraction layer rely on a combination of *universality*, which allow us to manage any web-accessible resource; *accessibility*, in terms of homogeneous and source-independent access to resources; *simplicity*, in terms of the general model and of the abstractions used, and *extensibility*, which is a property of both the model (which allow us to define different new resources and actions at different levels of abstractions) and of the architecture (that allow us to plug new resource managers).

The current status of this research work is a preliminary conceptual model of the resource space, and architecture and working prototype of the RSMS core (based on the resource management module of Gelee [7]).

As future work, we will define a detailed conceptual model of the scientific entities, and integrate it with the resource space conceptual model. We will extend the RSMS core to cope with advanced features such as dynamic adapter binding and distributed adapters, and once finished, we will bootstrap the platform with a set of common research services that we will derive from the liquid journals use case. It is worth mentioning that the platform described here will be validated by many other services developed within the Liquidpub project.

References

1. Casati, F., Giunchiglia, F., and Marchese, M. Liquid Publications: Scientific Publications Meet the Web. http://eprints.biblio.unitn.it/archive/00001313/01/073.pdf
2. Baez, M., Casati, F. Liquid Journals: Knowledge Dissemination in the Web Era. http://project.liquidpub.org/liquid-documents
3. Lossau, N. Search Engine Technology and Digital Libraries, Libraries Need to Discover the Academic Internet. D-Lib Magazine, 2004, Vol. 10, No. 6, ISSN 1082-9873. DOI= http://dx.doi.org/10.1045/june2004-lossau
4. Kahn, R.; Wilensky, R. A Framework for Distributed Digital Object Services; Corporation for National Research Initiatives, Architecture for Digital Library Research Project, 1995.
5. Franklin, M., Halevy, A., and Maier, D. 2005. From databases to dataspaces: a new abstraction for information management. *SIGMOD Rec.* 34, 4 (Dec. 2005), 27-33.
6. Halevy, A., Franklin, M., and Maier D. Principles of dataspace systems. PODS, 2006.
7. Baez, M., Casati, F., and Marchese, M. Universal Resource Lifecycle Management. ICDE/WISS 2009, Shanghai, China.