Resource Space Management Systems

Marcos Baez
Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento
Trento, Italy
baez@disi.unitn.it

Fabio Casati
Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento
Trento, Italy
casati@disi.unitn.it

Liquidpub[1] is an EU project within the "future and emerging technologies" category whose goal is to capture the lessons learned and opportunities provided by the Web and open source, agile software development to develop concepts, models, metrics, and *science support services* 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].

Novel services for science are a hot topic these days. From social bookmarking sites to online ranking of scientists, these services try to assist scientists in sharing content and assessing people and their scientific contributions. These services are however still very much anchored to a traditional notion of publication and are only scratching the surface of what can be done to help scientists collaborate for the greater good.

Examples of Scientific Services. An example of services that Liquidpub intends to deliver is that of *Liquid Journals*[1] (LJ), that redefines the traditional notion of journal which was born at a time where the *paper* was the only possible form of non-verbal knowledge dissemination, *printing* was the scarce resource, and therefore *peer review* and pre-publication filtering was necessary. Liquid journals are based on these notions i) separation of publication from inclusion in a journal: contributions are posted online (without any review) or published in traditional journals following a traditional process, and then they can be included in an arbitrarily high number of LJs. Each LJ decides policies and rules to determine if a contribution is included. Essentially, LJs are ways to aggregate all sort of available content based on what is interesting and relevant for its readers. This can be done via review, collaborative filtering, looking at journals of people we consider highly, etc; ii) Everybody (even individuals) can create and run LJs; iii) Papers are not the only source of knowledge. Blogs, experiments, datasets, slides, comments/feedback and the like are valid and useful forms of dissemination, some of them having the additional benefits of allowing early dissemination and therefore better collaboration. Including *feedback* as a form of contribution has the effect that it is considered as part of what is evaluated from a scientist and therefore it encourages giving feedback, which is fundamental to the scientific creation process.

All is driven towards what the purpose of a journal should be: providing people with interesting content to read, minimizing the dissemination overhead, and maximizing the collaboration. Current journals are a particular case of LJs. In terms of web services, liquid journals require an infrastructure that allows defining LJs and fetching/filtering content from the web based on profiles, preferences, recommendations, policies, and so on. The effort in developing the liquid journals is 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 scientific resources on the web, capable of merging results from various resource managers (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.

Another service LP provides is *research evaluation* (also based on LJs, but not only). Evaluation is a necessary aspect of research, not only to filter contributions but also to help select people for hiring or promotion. In this respect, the LiquidPub project aim at developing scientific metrics that i) take into account the different aspects of the research activity: that of creating content, filtering content, proposing good ideas, setting up good experiments, and ii) encourage "good" behaviors (sharing content early, providing feedback, etc) and that not only look at what people have done but that try to assess *interest* in what scientist will produce. Besides defining metrics, what we want to provide is a way to make it easy for scientists and evaluation agencies to define their own metrics. To this end, we need to provide services that allow programmatic access to scientific data and metadata -- both traditional ones (Google scholar, citeseer, citeUlike, SpringerLink,..) and more novel ones (blogs, liquid journals, …), that allows for sophisticated features such as author disambiguation or for comparing people of different communities and therefore having different scientific metrics (this is hard because it is hard to define what a community is), and that allow people to easily define and plug in their own metric which use data from their favorite sources.

Implications for Research Spaces Management Systems. Given the above, we need a common platform to access the various kinds of *scientific resources* available on the web, in a way that it easy (or at least easier) to develop services for scientists on top. For this, such a platform should provide *programmatic access* to scientific resources, hiding the tedious problem of accessing heterogeneous platforms which very often are not even available for programmatic access but are only designed for Web browser access (e.g., Google scholar).

The large (and growing) amount of scientific web applications providing access to these resources makes it practically impossible to design a monolithic infrastructure

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

that incorporates all of them. It is then required that such an infrastructure provides an extensibility facility that allows adding new services as needed.

We have also seen the need for a set of specific services in the examples above: services for extending the evaluation with user-defined metrics, primitives to manage author disambiguation, services for crawling various scientific metadata sites (e.g., for citations), services for observing resource usage (to provide recommendations), etc. To support applications like LJs, we need support for query that understands concepts such as relevance or interestingness, we need to be able to collect user feedback or observe users' actions if possible, and the like. We have also observed the need for a uniform conceptual model for scientific resources that is sufficiently general but also specific enough to be useful.

The previous observations led us to the design and development of a *resource space management system* (RSMS) for scientific resources. For this we borrow notions from the principles of *Dataspaces* [2] to apply it to a *space of scientific resources*. A resource is anything that has a URI, but the specific aspect is that RSMS is specifically focused on services to support knowledge dissemination. These resources are managed by potentially different service providers (e.g., Google Docs, Google scholar, ...). We refer to these service providers as *resource managers*. In a nutshell, the characteristics of the RSMS – and for all the applications we build on top – are:

- **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,** to cover the large set of scientific resources of various kinds of scientific resources as described above, not just papers.
- **Collaborative Extensibility,** to facilitate extensibility by the community where developers can just register scientific services. We bootstrapped the system with a few key access and crawling services, but the key is how to avoid overloading the system with hundred of adapters to access the different resource managers.

From the functional sides, the key is in understanding (and designing, implementing) which kind of actions are supported by the resource managers, which kind of horizontal services should be provided because they are useful to a large number of scientific services, and what is the underlying resource model to be exposed to the horizontal services as well as to the services to be developed on top.

In terms of models, 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. Resources can be *scientific contributions*, *people*, and *events,* and can be grouped (communities are groups of people, proceedings are groups of papers, conference series are groups of events).

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 Google-Docs regardless the differences in their "signature" detail.

Basic services provided by RSMS include among others support for scientific queries, which are queries that look for interestingness or relevance for a person or a contribution; crawling (e.g. to collect citation data from scholar); caching (e.g., caching of the crawled data); author disambiguation; and analytics, which observe usage and use this information for recommendation or similarity analysis (hence, in the query evaluation phase). With the access layer and the basic services (these and other available in RSMS) we can implement services such as LJs or research evaluation as well as many others, such as liquid books or lifecycle management.

In terms of extensibility, 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 services available without introducing changes into the platform. The resource manager and the concept of resource type collectively support static or dynamic binding to both adapters and (for services using the RSMS) to resources. Besides load balancing, the key benefit here is reliability and the ability to leverage the community to maintain a complex distributed system. 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.

We have implemented working prototypes of the RSMS and of services on top. These are available as open source from the project web page[1], along with a detailed description of the architecture and with instructions of how to collaborate. The *Liquid journals* tool is currently in early implementation phase, but with the code already available and open for collaboration.

ACKNOWLEDGMENT

This work has been supported by the EU ICT project LiquidPublication. The LIQUIDPUB project acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 213360.

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] 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.