

# Sharing Scientific Knowledge with Knowledge Spaces

Marcos Baez, Fabio Casati and Maurizio Marchese

Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, Italy  
{baez, casati, marchese}@disi.unitn.it

**Abstract.** This paper presents a set of models and an extensible social web platform (namely, Knowledge Spaces) that supports novel and agile social scientific dissemination processes. Knowledge Spaces is based on a model for structured, evolving, and multi-facet scientific resources that allows the representation of structured, evolving, and multi-facet scientific knowledge and meta-knowledge, of effective “viral” algorithms for helping scientists find the knowledge they need, and of interaction metaphors that facilitate its usage.

**Keywords:** knowledge dissemination, social web, scientific publications.

## 1 Introduction

Knowledge Spaces (kspaces for short) are a metaphor, a set of models and processes, and a social web platform that help you capture, share and find scientific knowledge, in all of its forms.

The principle behind kspaces is to allow knowledge dissemination in the scientific community to occur in a way similar to the way we share knowledge with our colleagues in informal settings. The rationale behind this is that when we interact informally with a small team of colleagues dissemination is very effective. We are free to choose the best format for communicating our *thoughts* and results, we share both established results as well as latest ideas, we interact and *carry on a conversation* (synchronously or via email), we *comment* on other people's contributions and papers and observe *relations* among various contributions. Even when we remain in the domain of papers, we often find that we come to know interesting papers not by doing a web search or scan the proceedings, but because we “*stumble upon*” them, that is, we have colleagues pointing them to us via email or mentioning them in a conversation (along with their comments). In other words *knowledge spreads virally*.

Kspaces aim at providing a set of models, processes, metrics and tools to support this informal, viral and social way of disseminating knowledge among the scientific community at large and via the Web, complementing the well-established method of papers published in conferences and journals after peer review. The goal is to use a web-based system to enable the capturing of these evolutionary bits of knowledge and data, however they may be expressed, as well as the capturing of ideas and opinions about knowledge, and leverage this information and meta-information to spread knowledge socially. Capturing opinions on knowledge is particularly important. The fact for example that somebody (and especially somebody we “trust”) shares a paper tells us a lot on the value of this paper, much more than a citation can do. As readers,

we relate specific papers, in our mind, with prior knowledge. When listening to a talk we think that other work is relevant to the one being presented and often we jot it down in our own personal notes. In a world where information comes out from the web like from a hose, this knowledge about knowledge becomes essential to dissemination. Tagging, annotating and connecting the dots (linking resources in a way much more useful to science than citations) become almost as important as the dots themselves.

Kspaces support this not only by using web technologies as the basis for its implementation but by using web 1.0 and 2.0 concepts in the way scientific resources and their relationships are modeled and in the way knowledge sharing is supported. In essence, kspaces is characterized by a conceptual model and a repository for scientific resources (or for pointers to them if stored elsewhere). Resources are linked in arbitrary ways and relationships are typed and can be annotated. This is analogous to the Web, although it is oriented to linking scientific resources and to supporting (and then leveraging) relationship types and annotations. Indeed building this evolving web of annotated resources and leveraging it to find knowledge is a key goal of kspaces. The intuition is that having such web of connected knowledge can be as instrumental or even *more* instrumental (because it contains more metadata) to finding knowledge than the Web is to finding web pages. Today this web of resources is simply not there and this is part of what makes finding interesting and relevant scientific knowledge hard.

On top of this space of resources, kspaces define specific processes, permissions, and interaction modes people use to share knowledge. Kspaces manifest themselves in various forms, called *designs*, tailored at capturing different forms of scientific knowledge shared in different ways, from maintaining a library of related work, talks, datasets, etc, in an area – including our own, evolving work - to forming knowledge communities, writing and publishing (liquid) books, supporting the collection of the knowledge that emerges in the brain of attendees during a talk, and many others. It is through spaces with specific design that knowledge and meta-knowledge is collected and disseminated. The dissemination and search of knowledge over kspaces is then based on the “social interest”, on the goals of a search (e.g., related work vs introductory material), and on the meta-knowledge (e.g., tags and annotations). Kspaces, although being richer and more flexible than many existing systems, is not the first and only platform that exploits some form of social meta-knowledge to support search. Mandeley, CiteUlike, and Connotea, just to name a few, all have some elements of this. We believe that the key to a successful platform here lies in how such meta-knowledge can be collected and how it is used, and here lies a key contribution of kspaces.

## 2 Knowledge Spaces

We see scientific contributions as a structured, evolving, and multi-facet objects. Specifically, we see the Scientific Resource Space (SRS) we want to collect, organize, share, evaluate, and search as consisting of scientific resources, organized as set of nodes in a graph, that can be connected and annotated by authors or readers. We do

not discuss or formalize here the SRS model further as it was discussed in our earlier work [1] to which we refer the reader for details.

A Knowledge Space is defined as  $KS=\{R, Q, M, Tr, C, S\}$ , i.e., a collection of SRS content, with the following characteristics:

- The content is defined intensionally (in terms of the properties the content should have) or extensionally (content is explicitly added). A space can be only intensional, only extensional, or a mix. In case the content is defined intentionally,  $KS$  defines in essence a query over the SRS, denoted as  $Q$ , while  $R$  denotes resources explicitly added.
- A  $KS$  has members  $M=\{O, E, V\}$  that can be owners  $O$ , editors  $E$ , and viewers  $V$ . Viewers can only access the resources. Editors can add or remove content. Owners are editors and can add new viewers or editors or owners.
- $Tr=\{transparent \mid opaque\}$  denotes the *transparency* flag. An *opaque* space is a space where the comments, tags, annotations on resources, and the existence of the space itself are only visible to the members of the space. In a *transparent* space, comments, tags, and the posted resources “percolate” down to the resource space. Non-members cannot see what’s in the space, but can see the tags and comments on the resources.
- $C=\{RST, RLT, ENT\}$  denotes the configuration of the space, i.e. a container for a specific  $KS$  application. Because containers are used for a purpose, they typically include specific types of resources and relationships that acquire a particular meaning, and require a specific UI representation.
- Spaces can also follow a *lifecycle* defined by a particular design: for instance in a implementation of a  $KS$  modeling panel discussions the space will go through the phases involving - at least - the *prior*, *during* and *post panel* discussions. At each stage  $S$  in the lifecycle, the permissions and the way the UI renders the content may differ.

A  $KS$  is itself a resource, and as such  $KS$  can be included in other  $KS$ s, it can be annotated and linked as resources do.

### 3 Knowledge Spaces Applications

Kspaces essentially are a general-purpose repository and a related API collection that can be used to develop applications for specific purposes around the area of collecting, linking, sharing, and finding scientific knowledge.

For example, a particular case of kspaces “Instant Communities” can reuse the kspace infrastructure and related API as foundations. Specifically, “Instant Communities” kspace provides an IT infrastructure that helps create a “community of interest” in real-time during the panel or session. *Initially*, material is created and posted before the panel, by the panelists. This is an immediate body of knowledge that can be shared among panelists and participants. Then, *during the panel*, attendees, while listening, if they have a tablet or laptop avail, they can add papers, comments, questions, slides, links, interesting datasets, and whatever they feel useful. *After the panel* the goal of Instant Communities is to facilitate collection and sharing of material, to keep the attendees in touch, and extend the community with other people interested. People can also create their own “view” on this body of knowledge, with a few clicks

and drag and drop. One can do so by explicit selection or by filtering by poster, topic, and the like.

They can then share this view, or the entire space, with their team at home, with colleagues, with the entire instant community, etc. Incidentally, all this adding, selecting, and sharing knowledge provides an implicit way to connect people, connect knowledge, and identify interesting knowledge (by looking at what people share). It is a way therefore to provide information that can be used for facilitating search and for assigning reputation to scientific resources.

The detailed list of features, user stories, screenshots and implementation details of instant communities are available at <http://open.instantcommunities.net>. The application has been used in various conferences and seminar series and will be deployed this fall for intra-company usage. It is one of the way in which kspaces tackle the challenges of bootstrapping and of usage: by providing knowledge capturing and sharing applications for specific purposes and communities.

## 4 Findings, Status and Next Steps

Kspaces is the result of several attempts and failures at arriving at a model for capturing knowledge, which we initially tackled by trying to impose a specific knowledge collection mechanism (that is, in our terminology, a single, specific kspaces application). The finding during the years of work on this tool is that, besides a proper conceptual model, we need very domain-specific and targeted applications if we want to lower the barriers to knowledge sharing based on the principles described in the introduction.

The concept and a preliminary implementation of kspaces, in their various forms and designs, are being exploited in several different pilots in cooperation with the EU Commission (who used it at their flagship event for future and emerging technologies, [fet11.eu](http://fet11.eu)), IEEE, Springer, the archeology museum in Cambridge and major international conferences to support the collection and sharing of knowledge in conferences, in technical communities, among scholars visiting museums, and in the generation of teaching material among groups of lectures.

## References

1. Baez, M., Birukou, A., Casati, F., Marchese, M. Addressing Information Overload in the Scientific Community. Addressing Information Overload in the Scientific Community," IEEE Internet Computing, vol. 14, no. 6, pp. 31-38, Nov./Dec. 2010, doi:10.1109/MIC.2010.107.

**Acknowledgements.** We acknowledge the great contributions to the ideas and the code from all the LiquidPub project team, with particular thanks to Alex Birukou and to all our kspaces developers including Delsi Ayala, Simone Dalcastagne, Nicola Dorigatti, Lyubov Kolosovska, Muhammad Imran, Michele Lunelli, Aliaksey Minyukovich, Daniil Mirilenko, Alejandro Mussi, Cristhian Parra, Laura Pop.

This work has been supported by the LiquidPub project (FP7 FET-Open 213360.).