Collaboratively building structured knowledge with DBin: from del.icio.us tags to an “RDFS Folksonomy"

Giovanni Tummarello
DERI Galway
National University of Ireland
+(353) 091 495285
g.tummarello@gmail.com

Christian Morbidoni
SEMEDIA
Università Politecnica delle Marche, Ancona, Italy
+(39) 071 2204841
c.morbidoni@deit.univpm.it

ABSTRACT
DBin is a Semantic Web application that enables groups of users with a common interest to cooperatively create semantically structured knowledge bases. These user groups, which we call “Semantic Web Communities”, are made possible by creating customized user environments called “Brainlets”. Brainlets provide user interfaces and domain specific tools (e.g. querying, viewing and editing facilities) which enable community participants to interact with the data of interest. Brainlets are directly created by domain experts using an XML description language. DBin clients communicate and exchange annotations using a P2P infrastructure. Access control and digital signatures put by DBin inside the authored RDF enable trust and information filtering. In this paper we show a specific use case where a “Semantic Web Community” is created to enable a group of users to share their del.icio.us tags and organize them into a cooperatively built RDFS ontology.

Keywords
Semantic Web, Tags, Ontology creation, DBin, peer-to-peer.

1. DBIN PLATFORM OVERVIEW
The DBin project is an integrated, end-user oriented, Semantic Web Platform. More in detail, it is a Semantic Personal Knowledge Manager (Semantic PKM) with the following main features:

- Based on the Semantic Web languages stack
- Topic independent, yet customizable to be domain specific.
- Ontology based reasoning used whenever possible for assisting the user (e.g. automatic rich user interface creation) in visualizing, editing and browsing data;
- Works as personal information manager and is run in a local desktop environment.
- Using a P2P algorithm, it can synchronize aspects of the local knowledge with that of other online DBin users.
- Is not a programmer toolkit. Most customizations can be done using XML scripting languages and ontologies.
- Rich client multiplatform software. Based on the Eclipse RCP, enjoys its plug-in system.

2. SEMANTIC WEB COMMUNITIES: THE USER EXPERIENCE
In this section we present the overall user interaction model as implemented by the DBin platform. Users might simply want to participate into Semantic Web communities (from here referred to as “regular” users) or might want to start up and/or maintaining them (power users). To participate means to be able to cooperatively build the community shared semantic knowledge. The power user starts up a new community by first creating a customized user environment for the editing and exploitation of semantically structured annotations. These environments are called Brainlets.

2.1 Brainlets
Brainlets [1], are plug-ins in the DBin platform (therefore, technically Eclipse Plug-ins) and can be though as “configuration packages” preparing the client application to operate on a specific domain (e.g. Wine lovers, Italian Opera fans etc.). From the user perspective, the relationship between Brainlets and the DBin platform is similar to that between HTML and a Web Browser. Much like HTML web sites, Brainlets are created in XML and RDF and do not require any programming skills. They customize aspects such as:

- The ontologies to be used for supporting knowledge creation and presentation of data;
- GUI Layout and coordination. Widgets are first “instantiated” from a rich set of predefined ones and then configured for the domain of interest, e.g., an ontology navigator might be configured to show certain classes or instances and to hide others. The components are then interlinked among each other; this means that chains of reactions to actions such as a focus change can be defined;
- Templates for domain specific annotations (e.g. a “Movie Brainlet” might have a “Review” template, with associated slots, that users can fill);
- Templates for readily available “pre-cooked” domain queries, which are structurally complex domain queries with only a few simple free parameters (e.g. “give me the name of the cinemas where a movie of genre X is being shown tonight”);
- A trust model and information filtering rules for the domain (e.g. public keys of well known “founding members” or authorities, preset “browsing levels”);
- Scripts for guiding the user in creating new URIs for domain resources (e.g. adding a new "paper" to the knowledge base);
- Scripts connected to Brainlet specific menus or buttons that implement domain specific functions;
- Support material, customized icons, help files etc.;

- Optionally Brainlets might contain support to Java code and libraries for add on capabilities beyond those provided by the standard Brainlet widgets;
- A basic RDF knowledge package.

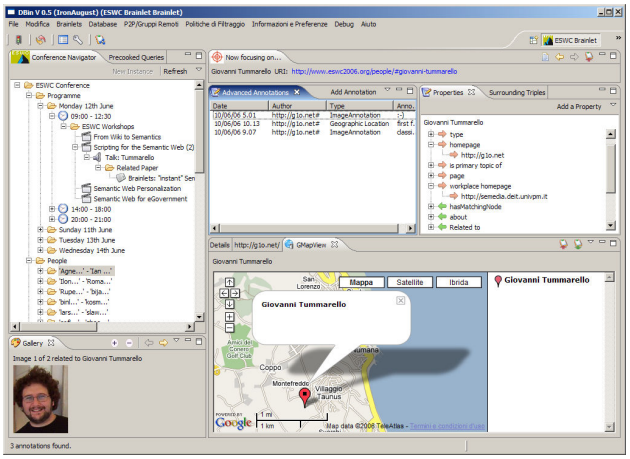

Figure 2. A Brainlet as experienced by an end user. The Semantic aware widget are positioned and made to interoperate by the Brainlet configuration.

To the end user, most of the above aspects are simply hidden behind the integrated Brainlets UI which presents itself, for example, as shown in Figure 1 (ESWC Budva Brainlet).

It is important to notice that the Brainlet UI is not simply a mash up of visualizers. As the components are coordinated among each other, the result is that a Brainlet guides the user into a meaningful and domain specific workflow interaction with the structured data. At any time, the domain ontologies are used as much as possible for assisting users in editing and browsing knowledge, for example to suggest which kind of annotations are possible for a given resource.

2.2 The overall scenario

Once Brainlets have been created by power users, they are installed by the regular users into their local DBin client.

Brainlets are downloadable files and as such they can be made available at a Web site by their creator. DBin itself, however, provides a mechanism for discovering new Brainlets as the user is browsing the P2P channels; as a user join a channel which was created for the users of a specific Brainlet, DBin will optionally guide the user to the Brainlet download and installation.

The overall scenario is depicted in Figure 3. On top of what has been illustrated in the previous section, Brainlets also have roles in how a user can connect to the others. In particular, a Brainlet contains pointers to P2P channels which are either known to contain information pertaining to the domain of interest or that the power user has previously created for this purpose. Creating a P2P channel for a specific topic is a simple operation that has to be performed on the configuration of an RDFGrowth server. RDFGrowth servers act as "meeting point" for the DBin clients but do not carry themselves metadata or binary attachments.

Binary attachments are stored by DBin automatically in a web accessible space. This is done by DBin interfacing with a web publishing system much similar to WebDAV[1] which we call "Data Publishing Service" (DPS). Unlike WebDAV, a DBin publishing service is a simple PHP script and, as such, it can be deployed with ease in most low cost commercial web hosting environments. For the end user convenience, the DBin platform comes with a default DPS setting[2]. The same Data Publishing mechanism provides the DBin users with the ability to create and publish RSS feeds and RDF dumps derived from the internal knowledge.

The Brainlet provides for a domain specific user interface as it instantiates and positions RDF aware widgets which are connected together to create an application workflow. It is important however to notice that they do not "take over" the individual installations; many Brainlets can coexist as needed.

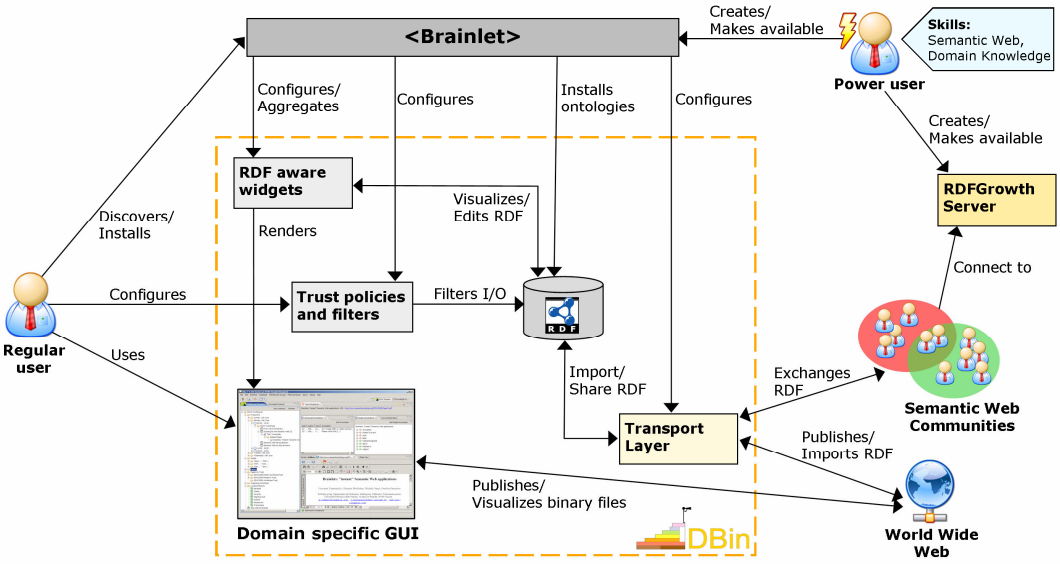

Figure 3 DBin and its relationship with different actors in the "Semantic Web Communities"

2.3 The RDFGrowth P2P algorithm

In this section we quickly overview the basic ideas and principles behind the RDFGrowth P2P metadata exchange algorithm, refer to [2] for a complete description of the algorithm.

Unlike previous approaches, which have explored P2P interactions among peers based on distributed queries, collecting and returning results, as in works like [3], [4], [5] and [6], RDFGrowth operates in a "greedy" and uncommitted scenario where cooperation between peers is minimal. It operates by direct queries that are in general of fixed computational cost. Without going into details, the algorithm provides synchronization of RDF knowledge among the user's DBin installations. Such synchronization is not performed in full, but along "aspects" of knowledge; it is restricted to those RDF triples which are very closely connected with a set of URIs defined "interesting" by a community "banner". The P2P community creator, usually the same person who created the Brainlet, defines an "URI interest banner", that we call Group URIs Exposing Definition (GUED), usually queries which have as a result a list of URIs. An example of GUED can be "select all resources of type *Papers* which have topic *Semantic Web*". Upon joining a community, a peer runs such queries to select the local set of resources about which knowledge will be synchronized with that of the other participants.

At user interaction level, DBin shows an interface that is somehow similar to that of popular file-sharing software. A list of servers is presented and, upon selecting one, the list of semantic P2P channels is displayed for the user to join. Furthermore, an access control mechanism allows for restricted P2P groups.

3. INTERACTION AMONG COMMUNITIES

It is interesting to see how multiple Semantic Web Communities relate both to each other and to the individual user.

Figure 4 shows a possible use case where each user participates in one or more communities with different topics of interest.

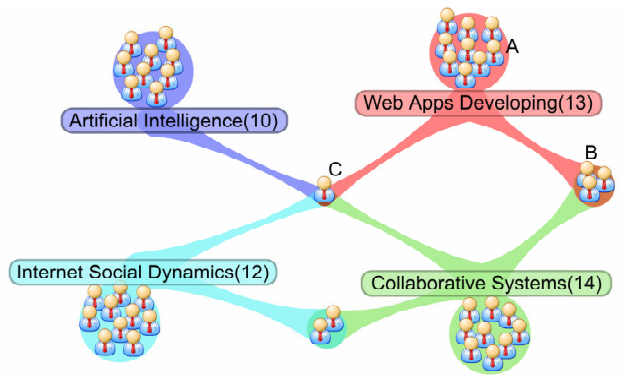

Figure 4 An example of users participating in multiple communities

Users in groups such as that of Alice (marked A in the illustration) are Web developers. Within their community the resources of interest are, for example, available web technologies and tools (such as PHP, Ajax, JSP, etc.). Participants in the community annotate such resources for example expressing opinions about the tools, pointing at web tutorials or at web sites that use specific technologies. On top of pure metadata, annotations, they can also point at rich media posted on the web (e.g. pictures, documents, long texts, etc.). Other users who receive such annotations in the group can then reply or further annotate each of these for their personal use or into public knowledge.

As mentioned earlier, the operator that selects which resources a client shares with the others is the GUED. A GUED for the Web development community might contain queries such as "all the resources of type *WebTechnology*", with respect to a specific ontology, chosen or developed by the community's creator, where the class *WebTechnology* is defined. Only the metadata involving resources that fit this definition of 'common interest' are made available by a peer to the others in the community. In this case such metadata would be for example statements like "Web site X uses web technology Y" or "Web page X deals with issues in using technology Y".

Users like Bob (marked B) have interests, which go beyond those of a single community. In this example Bob is interested in developing a collaborative tagging application, so he joins both the 'Web development' community and the 'collaborative systems' one, thus being able to import into his own DBin metadata coming from the two sources. At this point Bob is able to make joint queries across the two domains, e.g. "which are the technologies on which existing collaborative systems are based on". Finally, Carole (marked C), is a Semantic Web researcher, so she might decide to join all the communities as they all contain information which might be useful for her research activity.

The interconnection between Semantic Web Communities can be seen also under a second, very novel point of view. If Communities share identifiers (e.g. their own URLs for available web applications, URLs of their specification for web technologies) then an annotation (e.g. web site X is based on technology Y), originally posted in one community is automatically cross posted to the other community since the URI is of interest to both (belongs to the GUED of both groups). This aspect, to our opinion, represents a particularly novel feature of Semantic Web Communities as a communication mean. Information in fact flows across group boundaries when it is in fact relevant to the users participating in the different communities. This is opposed to what happens with traditional means such as mailing lists, web forums or newsgroups where information, arguably, has to be manually cross-posted.

4. THE DEL.ICIO.US BRAINLET

The tagging paradigm is increasingly been adopted by people for organizing web resources they visit. Systems like del.icio.us[3] allows to associate simple keywords to web resources while the user is navigating the Web. However, such applications only allow annotations to be a flat list of terms, while it would be obviously useful to organize them in taxonomies or establish relations among them and possibly with existing ontologies. In this section we illustrate the *del.icio.us Brainlet*, that deals with this issue.

To think of a specific use case let us consider a group of colleagues, each one using del.icio.us to tag web articles and resources of interest for their work. They also use a knowledge

[3] http://del.icio.us

management application (such as DBin) for cooperating and organizing internal documents. It is likely that a subset of the tags they created in their del.icio.us accounts will be conceptually related to or equivalent to some concepts present in the domain ontology. Using the DBin del.icio.us Brainlet it is possible to import lists of tags into the local RDF store, transform them into ontology classes and insert them in the class hierarchy.

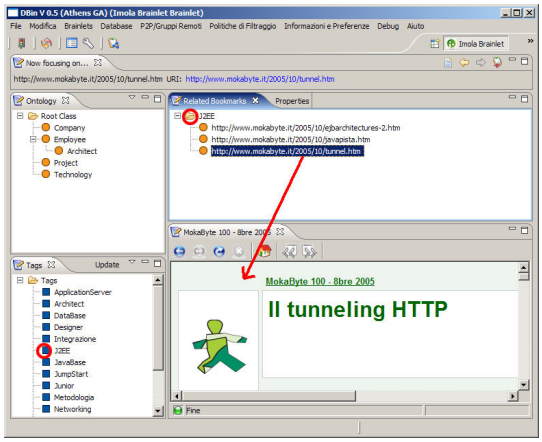

Figure 5. Upon selecting a tag the related bookmarks are listed and each of them can be visualized in the embedded browser.

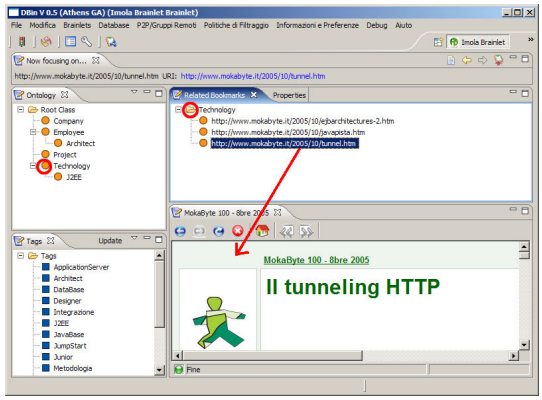

Figure 6. The JSEE tag has been identified as a sub-class of the Technology class, that automatically inherits the relation with the web resources tagged with JSEE.

By using the DBin P2P capabilities, such process is cooperative across the team. If necessary, DBin digital signature infrastructure would enable each team member to apply filters to see only contributions from certain members.

The screenshots shows this Brainlet in action. In Figure 5, the *ontology view* visualizes the taxonomy of the classes and provides functionalities to add new classes and subclasses as items of the tree, while the *tags view* shows the flat list of tags and gives the capability to update such a list from a del.icio.us account. Once a tag has been selected, Web pages which have been marked with that tag are listed in the *related bookmarks* view and the their content can be displayed in the *browser view*.

Upon selecting a tag (e.g. *J2EE*), a "transform into a sub-class" action is available to state that a tag is a sub-concept of a class in the ontology (e.g. *Technology*). This results in a new class being added to the ontology. As shown in Figure 6, when the user selects the class *Technology*, the web pages tagged with 'J2EE' are displayed in the right view, as such a tag has been stated to be a specification of the concept of technology.

The tags, as well as the pages and the other ontological terms, can then be annotates as any RDF resource in DBin. This enables annotations with comments, binary attachments, votes and any kind of structured annotation as defined by the Ontologies.

5. REFERENCES

[1] Tummarello, G., Morbidoni, C., Nucci, M. and Panzarino, O. Brainlets: "instant" Semantic Web applications. In Proceedings of the 2nd Workshop on Scripting for the Semantic Web at the European Semantic Web Conference (Budva, Montenegro, 2006)

[2] Tummarello, G., Morbidoni, C., Petersson, J., Puliti, P., Piazza, F. RDFGrowth, a P2P annotation exchange algorithm for scalable Semantic Web applications. 1st International Workshop on Peer-to-Peer and Knowledge Management (Boston, USA, 2004)

[3] Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M. and Risch, T. EDUTELLA: A P2P Networking Infrastructure Based on RDF. In Proceedings of the International World Wide Web Conference (Honolulu, Hawaii, 2002)

[4] Cai, M. and Frank, M. RDFPeers: A Scalable Distributed RDF Repository based on A Structured Peer-to-Peer Network. In Proceedings of the 13th International World Wide Web Conference (New York, USA, 2004)

[5] Nejdl, W., Siberski, W., Wolpers, M., Lser, A. and Bruckhorst, I. SuperPeer Based Routing and Clustering Strategies for RDF Based Peer-To-Peer Networks. In Proceedings of the 12th International World Wide Web Conference (Budapest, Hungary, 2003)

[6] Chirita, P. A., Idreos, S., Koubarakis, M. and Nejdl, W. Publish/Subscribe for RDF-based P2P Networks. In Proceedings of the 1st European Semantic Web Symposium (Heraklion, Greece, 2004)