ESWC 2006 Demo: DBin – enabling SW P2P communities

G.Tummarello, C.Morbidoni, M. Nucci, F.Piazza, P.Puliti - Università Politecnica delle Marche, Italy

http://semedia.deit.univpm.it – http://www.dbin.org

ABSTRACT

DBin is a general purpose, integrated, visually rich, open source, multiplatform Semantic Web that can be demonstrated and delivered to the end user today. With DBin, thanks to an integrated P2P engine, users can cooperatively annotate any domain of interest (under the methaphor of "group"). As individual users collect RDF from P2P groups and from any other sources, they are able to search and browse merged information in a maximally fast, rich and personalized way. DBin accommodates a number of modules to deal with specific issues ranging from visualization to trust.

1. Introduction

DBin is a user centered knowledge management platform revolving around a local, personal, Semantic Web Database. Content is inserted in this database in a number of ways:

- by a novel P2P Semantic Web algorithm (RDFGrowth) therefore fed from other DBin installations
- by specific modules integrating the content of the local machine (desktop integration).
- Explicitly by the users (which therefore contribute to the P2P knowledge)
- By the inclusion of external data sources or RDF graphs

All the knowledge stored in DBin is expressed using the languages defined in the Semantic Web initiative (RDF, RDFS) but the user doesn't necessarily have to be aware of this as the rich user interface will make it unnecessary to see or understand the basic information blocks.

2.Use scenario

A typical use of DBin might be similar to that of popular file sharing programs, the purpose however being completely different. While usual P2P applications "grow" the local availability of data, DBin grows RDF knowledge.

Once a user has selected the topic of interest and has connected to a semantic web P2P group, RDF annotations just start flowing in and out "piece by piece" in a scalable fashion. Such operations are clearly topic agnostic, but for the sake of the demonstration lets take an example of possible use of DBin by a Semantic Web researcher.

For example, a user who expresses interest in a particular topic and related papers (say "Semantic Web P2P") will keep a DBin open (possibly minimized) connected with a related P2P knowledge exchange group. He will then be able to review from time to time new pieces of relevant "information" that DBin collects from other participants. Such information might be pure metadata annotations (e.g. "the deadline for on-topic conference X has been set to Y") but also advanced annotations pointing at rich data posted on the web (pictures, documents, long texts, etc..). He could then reply or further annotate each of this incoming pieces of info either for his personal use or for public knowledge. If such replies include attachment data, DBin automatically takes care of the needed web publishing. At database level, all this information is coherently stored as RDF. At the user level however, the common operations and views are grouped in domain specific user interfaces, which in DBin are called "Brainlets".

ESWC 2006

3.Brainlets

Brainlets can be though of "configuration packages" preparing DBin to operate on a specific domain (e.g. Wine lovers, Italian Opera fans etc..). Given that Brainlet include customized user interface, the user might perceive them as full "*domain applications run inside DBin*" which can be installed as plug-ins and are suggested as soon as the user tries to enter a P2P group associated with the Brainlet itself. The message the user sees is similar to "The group you're trying to enter contains information which is best experienced with the X Brainlet, please visit page Y and install it". Continuing without said Brainlet is possible, but the interface wont be optimal for the given domain. In short Brainlets define settings for:

- The ontologies to be used for annotations in the domain
- A general GUI layout;. which components to visualize and how they are cascaded in terms of selection/reaction
- Templates for domain specific "annotations", e.g., a "Movie" brainlet might have a "review" template that users fill.
- Templates for readily available "pre cooked" domain queries.
- Templates for wizards which guide the user when inserting new domain elements (to avoid duplicated URIs etc)
- A suggested trust model and information filtering rules for the domain. e.g. Public keys of well known "founding members" or authorities,
- Basic RDF knowledge package for the domain

Creating Brainlets doesn't require programming skills, as it is just a matter of knowledge engineering (e.g. selecting the appropriate Ontologies) and editing of XML configuration files.

4.RDFGrowth: a scalable P2P engine based on the "minimum commitment" principle

The RDFGrowth algorithm powers DBin ability to collect RDF metadata from other peers with common interests. Previous projects, have explored P2P interactions among peers that rely on each other to forward query requests, collecting and returning results [3]. In contrast, RDFGrowth is designed to operate in a particularly "greedy" and

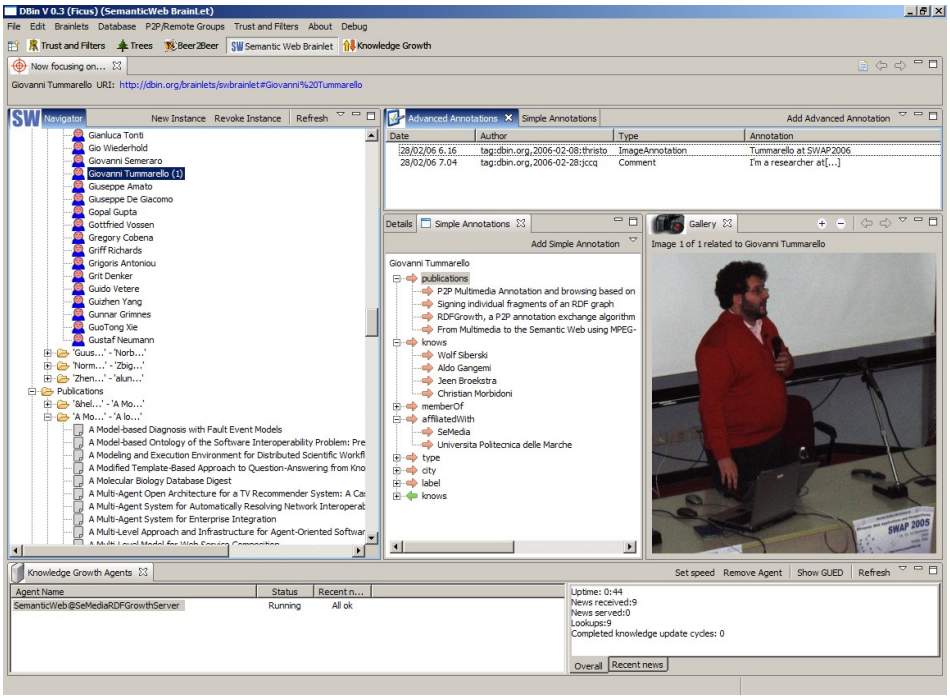

Figure 1 A screen shot of the SemanticWeb research Brainlet running. The principal "views" are: an ontology (and instances) browsing Navigator, a set of "Annotation" views and others related to searching, browsing, filtering etc. Kickstarting data is delivered insider the brainlet itself and has been adapted from that kindly made available by the Flink project [4].

uncommitted scenario where cooperation between peers is minimal. By this we mean that while peers are willing to provide some external service, the commitment should be minimal and in a "best effort" fashion. To obtain this, RDFGrowth follows a peculiar philosophy: minimum external burden.

- Given that a complex graph query could simply hog any machine, we assumed that individual peers would not, in general, be willing to answer arbitrary external queries. Any single peer would, if at all, answer just very basic ones. RDFGrowth only requires peers to answer very simple queries: basically the "RDF Surroundings" or blank node closure of the triples surrounding a specific URI. This type of query is not only very fast to execute but can also be cached very effectively.
- No "active information hunt" such as query routing, replication, collecting and merging is done. Such operations would require peers to do work on behalf of others that is again allowing peers to cause a potentially large external burden.

So, instead of querying around, in DBin a user browses only on a local and potentially very large metadata database, while the RDFGrowth algorithm "keeps it alive" by updating it in a sustainable, "best effort" fashion. A complete discussion is outside the scope of this introduction to the Demo, those interested can refer to [1] and other papers available from the DBin web site. As a result, keeping DBin open and connected to P2P groups with moderate traffic requires absolutely minimal network and computational resources.

5.Trust and URI Bridge Component

Due to the open nature of the P2P model (which can however be restricted to be used within organization or intranet), DBin also implements an RDF digital signature infrastructure that can be used by end users to perform custom trust based information filtering as well as signing annotations to be inserted in the system. For more details about the trust theory and infrastructure, see [2].

6.Conclusions and related works

DBin is an end user/power user centered application which provides an undoubtedly simplified, yet novel and exciting, all round and integrated Semantic Web experience. To the best of our knowledge there are no other projects which face the "all round" user scenario. Aspects of DBin capabilities can be directly compared with [5][6][7]. DBin is an Open Source project (GPL). Further documentation and compiled executables can be downloaded at http://dbin.org.

7.References

[1] G. Tummarello, C. Morbidoni, J. Petersson, Paolo Puliti, Francesco Piazza, "RDFGrowth, a P2P annotation exchange algorithm for scalable Semantic Web applications", 2004

[2] G. Tummarello, C. Morbidoni, P. Puliti, F. Piazza "Signing individual fragments of an RDF graph", WWW2005, poster track, Chiba

[3] W.Nejdl, B.Wolf "EDUTELLA: A P2P Networking Infrastructure Based on RDF" www2002 Honolulu

[4] Flink Project - http://prauw.cs.vu.nl:8080/flink/

[5] David Huynh, Stefano Mazzocchi, and David Karger. Piggy Bank: Experience the Semantic Web Inside Your Web Browser. International Semantic Web Conference (ISWC) 2005.

[6] P. Haase, J.Broekstra, M Ehrig, M. Menken, P. Mika, M. Plechawski. P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, C. Tempich "Bibster --- A Semantics-Based Bibliographic Peer-to-Peer System", ISWC2004

[7] Quan, Karger, "How to make a Semantic Web Browser" WWW2004