Put a Facebook button*, get the entire Web of Data to work for you

Giovanni Tummarello[1,2] and Szymon Danielczyk[1] and Michele Mostarda[2] and Davide Palmisano[2] and Renaud Delbru[1] and Stefan Decker[1]

[1] Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland
[2] Fondazione Bruno Kessler
Trento, Italy

Abstract. In this paper we introduce two Semantic Web based tools which helps website owners and their users with practical services. With just a bit of RDFa or Microformats, ordinary web pages can become contextually relevant, rich browsing experiences. Sindice Data Widgets and Site Search are such tools which uses this minimal markup to a) provide relevant information about the thing that the page is about, b) provide metadata based recommendation services across elements of the same website (without forcing the user to leave), c) provide rich faceted browsing functionality where users can browse the items offered by the website based also on the metadata obtained from external data sources.

1 Introduction: A brief history of Semantic Web data and exploiting applications

Despite considerable investments in research and many years since the standards have reached recommendation status, it has been difficult to demonstrate the benefits of the Semantic Web to end-users. It has also been challenging to show the benefits of the group of proposed methods and technologies over existing ones. Clearly, the problem has not been the lack of data since RDF has been made available for years at least in experimental ways from diverse sources. This became more and more the case with the advent of the Linked Open Data which, despite certain inherent complexities and limitations, managed to get to this day with hundreds of diverse dimensions and topics online as RDF data. At the time of this writing, some estimate that, over 20 billion triples exist in the wild. Despite this, however, RDF publishing remained somehow restricted to “semantic web partisan” environments and research funded projects while applications remained restricted to very few which often made use of just a few notable datasets (e.g., DBpedia) and almost never using web mechanisms or data flows resembling an open, and live flow of web data.

* or any other Web of Data markup really

Starting with 2008 however, certain attempts have been made by large web companies to somehow leverage semantic markup “out there”. In 2008, Yahoo! SearchMonkey team asked site owners and developers to mark up their pages with RDFa or Microformats, and gave them the ability to customize the appearance of their search results items. Despite the relative Yahoo! search market share, the mechanism had a clear appeal and purpose which caused a significant fraction of the web to be so marked up. The idea also appealed to Google which soon thereafter rolled out the Rich Snippet program basically offering similar functionality with the goal of providing more precise results to end-users. Despite Google’s cautious approach to enhancing the search page results with extra functionality or even modifications of the usual data layout, the Rich Snippet program is so far considered a success and is stably in production. While these platforms by Yahoo! and Google were important and leading in the right direction, the primary benefits laid outside of the web sites in question, under the control of these companies.

In 2010, Facebook introduced the OpenGraph protocol. Based on RDFa, Facebook OpenGraph reveals much richer annotations capabilities by containing several data types of common interest such as movies, books, actors, and songs to name a few, and useful common properties for events and locations. In return for such annotation task, as it only required a template update, Facebook provided the now very popular “I Like It” button which gave people the ability to vote for favourite items, paste them on ones own “wall”, and immediately see if friends “liked them too”. Despite OpenGraph’s expressiveness, a cold reaction was given by the Semantic Web community. Again, while the data was indeed a bit better than before, it was not considered to be as interesting, given that more appropriate and detailed data sets existed elsewhere (e.g., Geonames for physical locations, DBpedia for conceivably on every topic)

With these experiences in perspective, our work was conceived from the following idea: could it be that the interesting thing about data was not the data itself, but the small contextual hint it provides about the page (and collectively, all the annotations about the site)?

2 Toward Sindice Site Services: Required Ingredients

The Sindice Site Services we will introduce in the next section are Data Widgets and Site Search. Although they offer different independent features to site owners and end-users, the underlying principles are similar. More precisely, they both require the following fundamental ingredients: Effective data acquisition and a global search API.

2.1 Effective Data Acquisition

It has been a few months that Sindice offers what we call “Efficient Data Discovery and Sync”[3]. It allows sites that offer RDFa and Sitemaps to be indexed

[3] http://blog.sindice.com/2010/07/09/sindice-now-supports-efficient-data-discovery-and-sync/

effectively; where there will be reasonable expectations that they will be 100% crawled with politeness and effectiveness and they will be kept in sync with a delay shorter than 24 hours (in the current version). For example, sites like scribd.com support RDFa and have Sitemaps and therefore provide Sindice with thousands of updates per week, all of which are quickly made available in the index and in the internal caches for Sindice data services. To activate Sindice effective data acquisition, all that is needed is a single ping through the Sindice Ping interface. If the site is not previously "known" by Sindice, it will be analyzed and then if a Sitemap pointing to RDFa is available it will be used from then on.

It is therefore safe to say that under these conditions any website offering an "I Like it" button and complying to the above mechanism button can be effectively acquired and queried by Sindice.

2.2 A global search API endpoint

Once data is acquired from several websites, it must be indexed so that pieces from different websites can be mashed up together. This has been possible in Sindice for some time, as demonstrated in applications like Sig.ma (a winner at the Semantic Web Challenge 2009) which used a combination of Sindice's high performance document oriented APIs (the basic Sindice search capabilities) and of the Sindice cache APIs to provide custom, entity oriented mashups coming from dozens of automatically selected open data sources. Since then however, Sindice internally offers a full featured, quasi real time-synchronized SPARQL endpoint based on the the integration between Sindice and the Virtuoso RDF database, cluster edition. It is the full power of SPARQL in terms of being able to join bits and pieces across graphs, along with the very useful extensions provided by the vendor, that is going to be used in the two distinct applications.

3 Introducing Sindice Site Services - A Case Study: the world most unfortunate DVD rental website

To illustrate the Sindice Site Services we present here the following use case. Lets consider the world's most unfortunate DVD rental website. The site is so poor that it only has a simple database table with only the title of the movie. A possible page presentation for our website could be as in Figure 1.

Not surprisingly, the website owners are not so happy about the site. It has the most basic functionality, but it is neither appealing or fosters good navigation. They then decide to add the ubiquitous Facebook "I Like It" button. This way they get social functions; such as a count of people liking it (with a list of the user's "friends" that did that) as well as added advertisement when someone presses the button from their website. The page then looks as follows:

In doing so, they have - without realizing - joined the Web of Data by marking up their pages with some very simple markup.

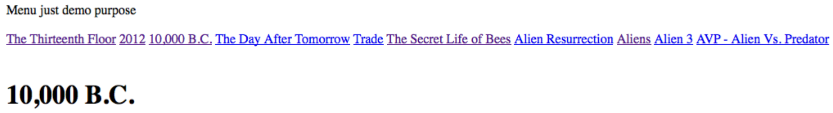
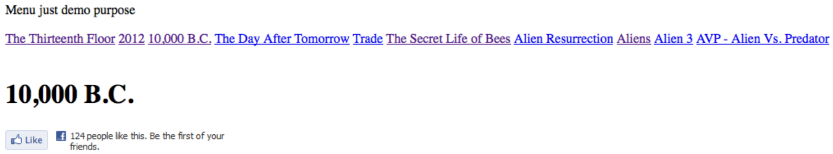

Menu just demo purpose
The Thirteenth Floor 2012 10,000 B.C. The Day After Tomorrow Trade The Secret Life of Bees Alien Resurrection Aliens Alien 3 AVP - Alien Vs. Predator
10,000 B.C.

Fig. 1: An example page from the world most unfortunate DVD rental website. Just the title of the movie (i.e. ”10,000 BC”) is available in the database and shown to the user


Menu just demo purpose
The Thirteenth Floor 2012 10,000 B.C. The Day After Tomorrow Trade The Secret Life of Bees Alien Resurrection Aliens Alien 3 AVP - Alien Vs. Predator
10,000 B.C.
Like

124 people like this. Be the first of your friends.

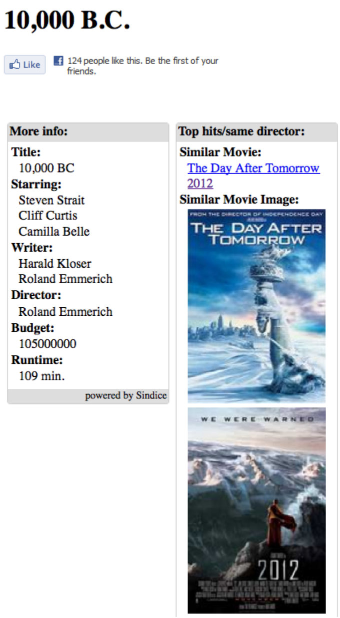
Fig. 2: the same site now supports a Facebook “I Like It” button, therefore exposing in RDFa at least the object type “movie” and title

3.1 ...enter Sindice Data Widgets

Sindice Data Widgets are small JavaScript tags which are meant to be placed in web pages which have semantic annotations. Thanks to the annotations on the page, the widget have the “context” of the page (e.g. the title of the movie being there displayed) and can provide information and services based both on any piece of data coming from the same website (as Sindice can be assumed to know about the entire site) but also on data coming from any combination of other sites on the Web of Data. The owner of the previously described site only has to 1) reach the Sindice Data Widget homepage, select from a list of widgets about given datatypes (e.g., OpenGraph ones like “Movies” or “Books”) and 2) copy-paste the JavaScript into the site HTML template. With this simple action, all the site pages about movies can now show widgets as those in Figure 3.

For each kind of data multiple widgets can be available e.g. in our examples “More infos about a movies” and “movie recommendation from the same Director” are shown, based on data joined from DBpdia and RottenTomatoes. Possibly a very notable feature of the mechanism is that it makes it simple to create a widget, e.g. like the one shown to the right - the one with movies from the same directors will show elements only if they are listed on the same site which embeds the widget. In this example, those are not all the movies from the same director, but only those that are available in our very unfortunate DVD rental website. Technically this is again possible assuming full site knowledge and via a join which uses the complete graph of the embedding widget. Any link in the widget will therefore not point to Wikipedia or any external abstract page, but simply to the website’s internal pages.

The widget mechanism is of course completely generic in terms of data: widgets can be created about any topic and joining from virtually any site from the Web of Data, thanks to the SPARQL query which operate on the entire



10,000 B.C.

Like | 124 people like this. Be the first of your friends.

More info:

Title:
10,000 BC
Starring:
Steven Strait
Cliff Curtis
Camilla Belle
Writer:
Harald Kloser
Roland Emmerich
Director:
Roland Emmerich
Budget:
105000000
Runtime:
109 min.

powered by Sindice

Top hits/same director:

Similar Movie:
The Day After Tomorrow
2012
Similar Movie Image:

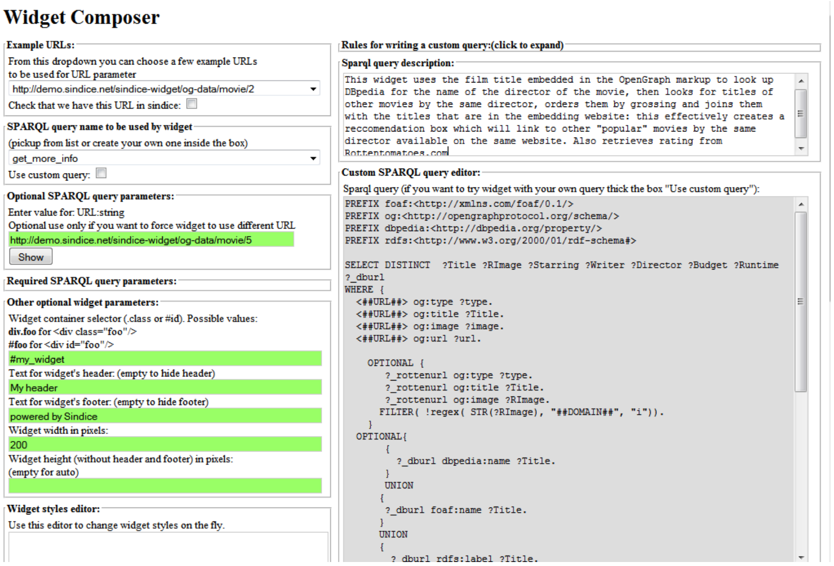
Fig. 3: The same site when the Sindice widgets JavaScript is copy-pasted into the HTML. The two widgets are shown here; a "Movie info" widget and a "Movie Recommendation" widget based on metadata from DBpedia

Semantic Web (as copied inside Sindice) as a single big quad store. In fact a key point of the Sindice Data Widget is that, widgets can be created by data savvy end-users using the provided widget editor, as depicted in Figure 4. Once created, the widget goes through a public approval phase, where they're tested not to cause unrealistic SPARQL resource drain, and then possibly listed as public widgets for others to embed or use as starting point for customized ones.

3.2 ... enters Sindice Site Search

Back at the world's most unfortunate DVD rental website, while the widget provided improvements and increased page views by providing interesting navigational links, they realize that their current search facility is quite limited: movies can only be searched by title, as it is the only field in their database. They are now about to discover, however, that adding those two triples of OpenGraph metadata was indeed going to be a pretty fruitful investment.

Given that Sindice has the entire content of the site, it can use all the extra metadata generated by the widget (like) queries to provide a full featured faceted search engine to locate and browse through the content of the entire website. This means that it will be able to browse and restrict searches also - or in this case almost entirely - based on metadata which is not available in the original website's database. For example, in Figure 5 the DVD offerings of the site can be



Widget Composer

Example URLs:

From this dropdown you can choose a few example URLs
to be used for URL parameter

http://demo.sindice.net/sindice-widget/og-data/movie/2

Check that we have this URL in sindice: □

SPARQL query name to be used by widget

(pickup from list or create your own one inside the box)

get_more_info

Use custom query: □

Optional SPARQL query parameters:

Enter value for: URL:string
Optional use only if you want to force widget to use different URL

http://demo.sindice.net/sindice-widget/og-data/movie/5

Show

Required SPARQL query parameters:

Other optional widget parameters:

Widget container selector (.class or #id). Possible values:
div.foo for <div class="foo"/>
#foo for <div id="foo"/>

#my_widget

Text for widget's header: (empty to hide header)

My header

Text for widget's footer: (empty to hide footer)

powered by Sindice

Widget width in pixels:

200

Widget height (without header and footer) in pixels:
(empty for auto)

Widget styles editor:

Use this editor to change widget styles on the fly.

Rules for writing a custom query:(click to expand)

Sparql query description:

```
This widget uses the film title embedded in the OpenGraph markup to look up
DBpedia for the name of the director of the movie, then looks for titles of
other movies by the same director, orders them by grossing and joins them
with the titles that are in the embedding website: this effectively creates a
reccomendation box which will link to other "popular" movies by the same
director available on the same website. Also retrieves rating from
Rottentomatoes.com
```

Custom SPARQL query editor:

Sparql query (if you want to try widget with your own query thick the box "Use custom query"):

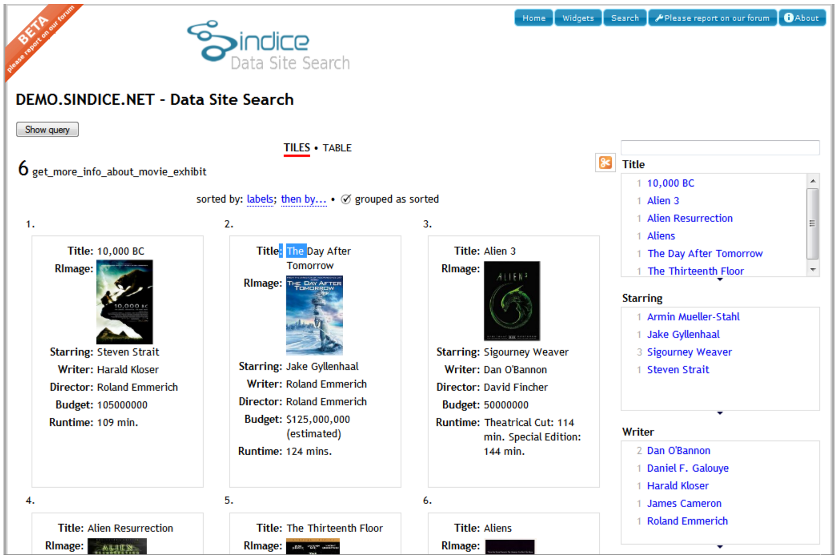
```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
PREFIX og:<http://opengraphprotocol.org/schema/>
PREFIX dbpedia:<http://dbpedia.org/property/>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT  ?Title ?RImage ?Starring ?Writer ?Director ?Budget ?Runtime
?_dburl
WHERE {
  <##URL##> og:type ?type.
  <##URL##> og:title ?Title.
  <##URL##> og:image ?image.
  <##URL##> og:url ?url.

    OPTIONAL {
       ?_rottenurl og:type ?type.
       ?_rottenurl og:title ?Title.
       ?_rottenurl og:image ?RImage.
      FILTER( !regex( STR(?RImage), "##DOMAIN##", "i")).
    }
  OPTIONAL{
       {
         ?_dburl dbpedia:name ?Title.
       }
       UNION
      {
       ?_dburl foaf:name ?Title.
      }
      UNION
      {
        ?_dburl rdfs:label ?Title.
```

Fig. 4: An early version of the widget composer. A SPARQL query is at the core of the widgets. Facilities are provided for testing the widget on example data, working with external parameters and to foster social exchange and reuse of widgets

restricted by director, writer, budget and possibly any other related metadata based on the same sources as before. As for the widgets, any link in the site search points by default to the specific web pages belonging to the site using the service. The site search service is provided at sindice.com/sitesearch/sitename can be embedded transparently in the DVD site. Their search engine is automatically updated as the Sindice crawler discovers more or different metadata using the efficient data synchronization capabilities previously highlighted.

4 Conclusion

We have presented two distinct applications enabled by the Sindice “Site Services” model. The simple but novel idea here is that apparently shallow markup can indeed have a key role not as a dataset per se, but as a target and ”magnet” for context specific data mashups and services. Semantic Web technologies here provide a crucial and clear benefit: SPARQL operating over a huge ”named graph” model allows data operations and mashups encompassing any number of diverse sources in a coherent way, something very different from standard API based web 2.0 mashups. The simplicity of RDFa markup makes it so that any website can become a data source to this model and have clear benefits (e.g. back links) in providing their data. Sindice Services are currently in alpha stage, with a public beta likely available in November 2010.



DEMO.SINDICE.NET - Data Site Search

6 get_more_info_about_movie_exhibit

sorted by: labels; then by... • ☑ grouped as sorted

1.

Title: 10,000 BC
RImage:
Starring: Steven Strait
Writer: Harald Kloser
Director: Roland Emmerich
Budget: 105000000
Runtime: 109 min.

2.

Title: The Day After Tomorrow
RImage:
Starring: Jake Gyllenhaal
Writer: Roland Emmerich
Director: Roland Emmerich
Budget: $125,000,000 (estimated)
Runtime: 124 mins.

3.

Title: Alien 3
RImage:
Starring: Sigourney Weaver
Writer: Dan O'Bannon
Director: David Fincher
Budget: 50000000
Runtime: Theatrical Cut: 114 min. Special Edition: 144 min.

4.

Title: Alien Resurrection
RImage:

5.

Title: The Thirteenth Floor
RImage:

6.

Title: Aliens
RImage:

Title

1	10,000 BC
1	Alien 3
1	Alien Resurrection
1	Aliens
1	The Day After Tomorrow
1	The Thirteenth Floor

Starring

1	Armin Mueller-Stahl
1	Jake Gyllenhaal
3	Sigourney Weaver
1	Steven Strait

Writer

2	Dan O'Bannon
1	Daniel F. Galouye
1	Harald Kloser
1	James Cameron
1	Roland Emmerich

Fig. 5: The same site now can support an advanced faceted browser for its DVD collection based almost entirely on data coming from external sources. The site search is hosted at sindice.com/sitesearch/sitename or can be embedded via JavaScript in any site page

5 Appendix: How we meet the SWC challenge criteria

Minimal Requirements: The application(s) is indeed an end-user application as it provides value to the general Web user and to website owners (1). The information sources used are (2) diverse and under diverse ownership or control are heterogeneous and contain very substantial amounts of data. In fact its fascinating to notice that one could create useful widgets or site search by deeply exploring Sindice for new and unknown valuable datasources (there are tricks for doing this). (3) Semantic Web technologies are entirely used, manipulating deeply data at query level, at presentation level, doing joins across datasets. Alternate technologies would be horrendously less efficient at doing this; consider hand-coding custom programming mashups and scraping data or having to study different API over time.

Additional Requirements: (1) The application provides an attractive and functional web interface to both end-users and site owners. The application will scale as much as the underlying cluster technologies (we're reasonably confident about that). There are all sorts of caching and batch calculations, and other Web 2.0 tricks that can be applied if needed (2). The application does potentially use all of the data that is currently published on the Semantic Web (3). Rigorous evaluations has not taken place to demonstrate the benefits of semantic technologies, or validate the results obtained - in this phase we trust on the good judgement of the reader (4). The dataflow here illustrated is novel to the best of our knowledge (5). Functionality and possible applications are multiple here and possibly beyond of what we can imagine now (6). We believe these services have a clear commercial potential (7), contextual information is crucial (8), some multimedia is used; for example, see the widgets example featuring images, videos could as well be embedded (9), parameters injected in queries allow dynamic workflow for widgets (10), the results can be pretty accurate indeed (11) if not ask the data producers to improve their data, finally a reason for them to produce useful data. (12) HTML should be accessible on multiple devices. Certain datasources might offer data in multiple languages (e.g. DBpedia), it would be a relatively simple addition to support this.

6 Acknowledgements

This work is supported by Science Foundation Ireland under Grant No. SFI /02/ CE1/I131 "Lion 2" and the LOD2 FP7 project No. 257943 http://lod2.eu/ . Gratitude also goes to Sarven Capadisli for supporting activity.