Sig.ma: Live Views on the Web of Data[*]

Giovanni Tummarello
DERI, National University of Ireland, Galway
FBK, Trento, Italy

Richard Cyganiak
DERI, National University of Ireland, Galway

Michele Catasta[†]
Ecole Polytechnique Federale de Lausanne (EPFL)
Lausanne, Switzerland

Szymon Danielczyk
DERI, National University of Ireland, Galway

Renaud Delbru
DERI, National University of Ireland, Galway

Stefan Decker
DERI, National University of Ireland, Galway

ABSTRACT

We demonstrate Sig.ma[1], both a service and an end user application to access the Web of Data as an integrated information space. Sig.ma uses an holistic approach in which large scale semantic web indexing, logic reasoning, data aggregation heuristics, ad hoc ontology consolidation, external services and responsive user interaction all play together to create rich entity descriptions. These consolidated entity descriptions then form the base for embeddable data mashups, machine oriented services as well as data browsing services. Finally, we discuss Sig.ma's peculiar characteristics and report on lessions learned and ideas it inspires.

Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Information Storage and Retrieval—*On-line Information Services*; H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*; H.4.3 [**Information Systems**]: Information Systems Applications—*Communications Applications*

General Terms

Algorithms, Experimentation

Keywords

aggregated search, RDFa, semantic web, web of data

1. INTRODUCTION

The amount of Resource Description Framework (RDF) documents and Microformats available online (e.g. in RDFa or published using the Linked Data approach) has grown significantly in the past years but yet there is still a strong need to demonstrate convincing applications that can exploit multiple, distributed, data sources when solving a task of interest to the user. The task at hand is however particularly complex. Assuming that an entity is indeed sufficiently described by available Semantic Web data, these descriptions can often be very heterogeneous and exhibit problems such as different describing ontologies, missing links between descriptions, little or no reuse of identifiers for the same entity, data errors, poor RDF publishing practices, and more.

In this paper, which extends [3], we present Sig.ma, an approach to Semantic Web Data consolidation which makes a combined use of Semantic Web querying, rules, machine learning and user interaction to effectively operate in real-world semantic web data conditions. As a result of this, Sig.ma provides the following end user services:

Advanced Browsing the Web of Data. Starting from a textual search, the user is presented with a rich aggregate of information about the entity likely identified with the query (e.g., a person when the input is a person name). As the user visualizes the aggregate information about the entity, links can be followed to information about related entities.

Live views on the Web of Data: rich, embeddable, addressable. At any aggregation page, Sig.ma offers rich interaction tools to *expand* and *refine* the information sources that are currently in use as well as some *data oriented* clean-up functionalities to *hide* and *reorder* values and properties. As a result, users can interactively create curated "views" on the Web of Data about a given entity which can be then addressed with persistent URLs, therefore passed in IMs or emails, or embedded in external HTML pages. These views are "Live" and cannot be spammed: new data will appear on these views exclusively coming from the sources that the mashup creator has selected at creation time.

Structured property search for multiple entities, Sig.ma APIs. A user, but more interestingly an application, can make a request to Sig.ma for a list of properties and a list of entities. For example requesting "affiliation, picture, email, telephone number, [...] @ Giovanni Tummarello, Michele Catasta [...]" Sig.ma will do its best to find the specified properties and return an array (raw JSON or in a rendered page) with the requested values.

[*]The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and in part by the FP7 EU Large-scale Integrating Project OKKAM - Enabling a Web of Entities (contract no. ICT-215032).

[†]This work was done while the author was at DERI

[1]`http://sig.ma/`

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

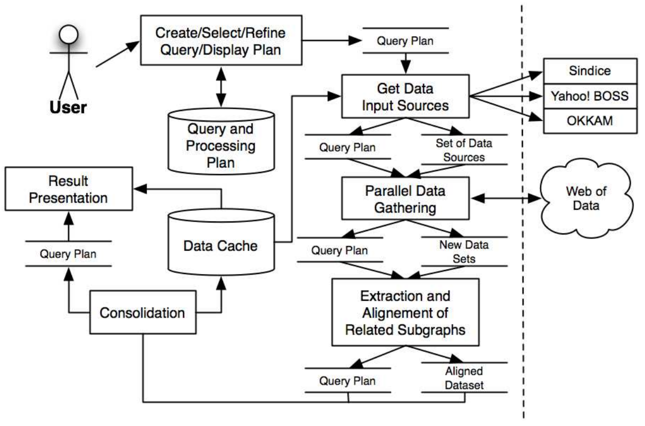


Figure 2: Sig.ma dataflow

2. TEST DRIVING SIG.MA: EXAMPLE USER INTERACTIONS

Before discussing the internals, it is useful to see how Sig.ma presents itself to the end user trough some typical interactions. We also encourage the reader to try the system online.

2.1 Sig.ma: Axel Polleres

In case of researcher “Axel Polleres”, plenty of data sources are available: RDF sources such as DBLP, Ontoworld, Semanticweb.org but also Microformat sources such as Polleres' public Facebook and LinkedIn profiles which, for instance, add more pictures to the mashup. Particularly rich sources such as the RDF coming from the DERI institute team page[2] add data such as his work phone number, publications and related projects. As ambiguity on the name is low, pressing “Add More Info” button returns many more relevant results which provide social contacts, alternative affiliations from previous employers and more. The result of an aggregation of 30 sources is shown in Fig. 1.

3. SIG.MA: PROCESSING DATAFLOW

Sig.ma revolves around the creation of *Entity Profiles*. An entity profile – which in the Sig.ma dataflow is represented by the “data cache” storage (Figure 2) – is a summary of an entity that is presented to the user in a visual interface, or which can be returned by the API as a rich JSON object or a RDF document. Entity profiles usually include information that is aggregated from more than one Web source. The process of creating an entity profile involves several steps which are describedin the next sections.

3.1 Creation of a Sig.ma query plan

The process of creating a Sig.ma query plan takes three inputs, each of which is optional: A keyword search phrase, a number of source URLs, a number of resource identifiers (URIs). The difference between the last two items is that a source URL names a document, which is accessible on the Web, and might contain descriptions of any number of entities. A resource identifier names a specific entity, but may or may not be resolvable to a web document.

[2]`http://www.deri.ie/about/team/`

The initial user interface shown to a Sig.ma user presents an input box that allows entry of either a search phrase, or a single resource identifier. Other combinations of inputs are accessed through hyperlinks either from within Sig.ma or from a permalink.

3.2 Data Source selection and parallel fetching

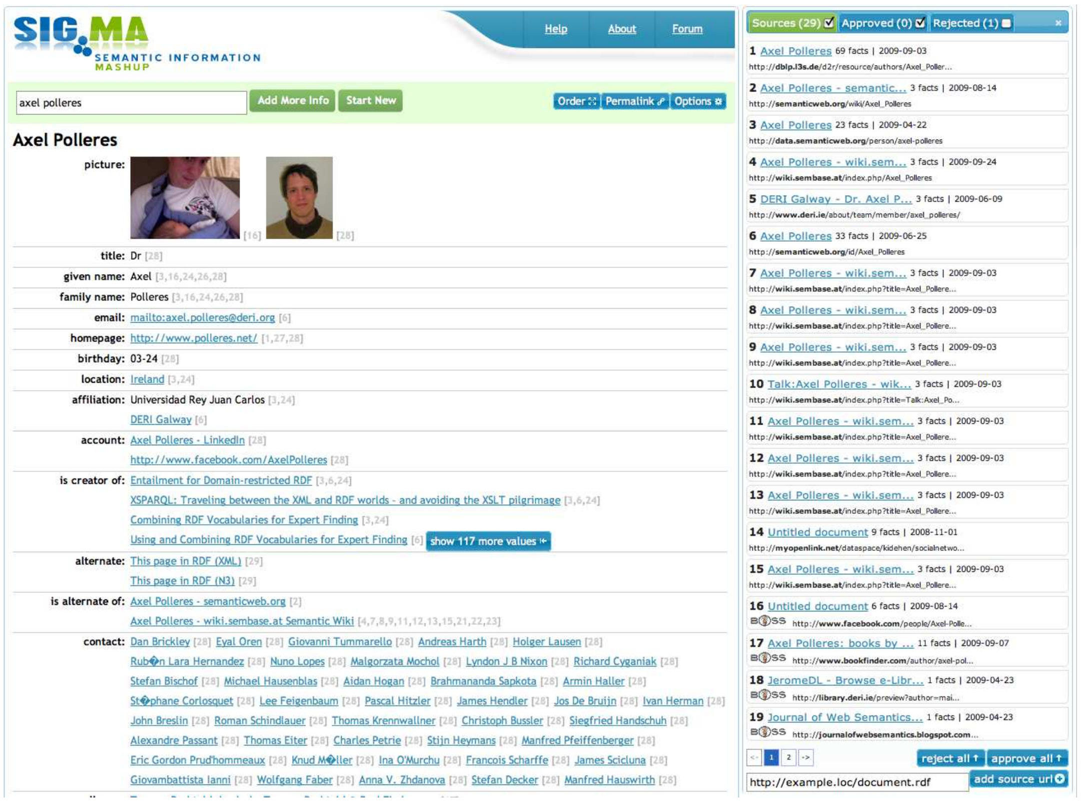
The first challenge is to identify a set of initial sources that describe the entity sought for by the user. This is performed via textual or URI search on the Sindice index and yields a set of of source URLs that are added to the input source URL set. The Sindice index does not only allow search for keywords, but also for URIs mentioned in documents. This allows us to find documents that mention a certain identifier, and thus are likely to contribute useful structured information to the description of the entity named by the identifier. Then, we interleave these results with the candidate list returned by the Yahoo! BOSS API[3], that we process to fit our peculiar scenario: basically, we consider the given URL to be interesting if and only if their metadata extraction layer detected semi–structured content in the page. The starting mashup is performed using the first 20 sources, but the user interface has then a control for requesting more resources. Sources are then fetched in parallel in a process mediated by multiple cache levels, e.g., making ample use of the Sindice public cache. The open source Sindice *any23*[4] parser is used to extract RDF data from many different formats.

3.3 Extraction and Alignment of related subgraphs

The structured RDF graph extracted from each source is broken down into chunks (called *resource descriptions*) that each describe distinct entities. A resource description contains the outgoing and incoming RDF triples of a specific resource together with other triples generated via transformation when specific cases are detected. As an example of a decomposition into resource descriptions, consider the case of a typical FOAF[5] file that describes a person. It will be decomposed into one resource description for the file's owner, one small description for each of their friends listed in the profile, and possibly one description for the FOAF document itself, containing statements about its `foaf:maker` and `foaf:primaryTopic`.

Resource descriptions are now ranked. A resource that has one of the resource identifiers from the source acquisition step will receive a large boost, as there is near-certainty that it describes the entity in question. Each description will be matched and scored against the keyword phrase, considering both RDF literals and (with a lower score) words in URIs. This helps to pick out the correct resource in cases such as FOAF files, which talk about multiple people, but it is easy to select the right one given a name. Resource descriptions below a certain threshold are removed from consideration. We now have a ranked list of descriptions that are hoped to describe the same entity. Since fuzzy keyword matching is used in several places in the process, the result is still subject to false positives.

[3]`http://developer.yahoo.com/search/boss/`
[4]`http://developers.any23.org/`
[5]`http://www.foaf-project.org/`



axel polleres

Axel Polleres

picture:	[16] [28]
title:	Dr [28]
given name:	Axel [3,16,24,26,28]
family name:	Polleres [3,16,24,26,28]
email:	mailto:axel.polleres@deri.org [6]
homepage:	http://www.polleres.net/ [1,27,28]
birthday:	03-24 [28]
location:	Ireland [3,24]
affiliation:	Universidad Rey Juan Carlos [3,24]
	DERI Galway [6]
account:	Axel Polleres - LinkedIn [28]
	http://www.facebook.com/AxelPolleres [28]
is creator of:	Entailment for Domain-restricted RDF [3,6,24]
	XSPARQL: Traveling between the XML and RDF worlds - and avoiding the XSLT pilgrimage [3,6,24]
	Combining RDF Vocabularies for Expert Finding [3,24]
	Using and Combining RDF Vocabularies for Expert Finding [6]
alternate:	This page in RDF (XML) [29]
	This page in RDF (N3) [29]
is alternate of:	Axel Polleres - semanticweb.org [2]
	Axel Polleres - wiki.sembase.at Semantic Wiki [4,7,8,9,11,12,13,15,21,22,23]
contact:	Dan Brickley [28] Eyal Oren [28] Giovanni Tummarello [28] Andreas Harth [28] Holger Lausen [28]
	Rub�n Lara Hernandez [28] Nuno Lopes [28] Malgorzata Mochol [28] Lyndon J B Nixon [28] Richard Cyganiak [28]
	Stefan Bischof [28] Michael Hausenblas [28] Aidan Hogan [28] Brahmananda Sapkota [28] Armin Haller [28]
	St�phane Corlosquet [28] Lee Feigenbaum [28] Pascal Hitzler [28] James Hendler [28] Jos De Bruijn [28] Ivan Herman [28]
	John Breslin [28] Roman Schindlauer [28] Thomas Krennwallner [28] Christoph Bussler [28] Siegfried Handschuh [28]
	Alexandre Passant [28] Thomas Eiter [28] Charles Petrie [28] Stijn Heymans [28] Manfred Pfeiffenberger [28]
	Eric Gordon Prud'hommeaux [28] Knud M�ller [28] Ina O'Murchu [28] Francois Scharffe [28] James Scicluna [28]
	Giovambattista Ianni [28] Wolfgang Faber [28] Anna V. Zhdanova [28] Stefan Decker [28] Manfred Hauswirth [28]

1 Axel Polleres 69 facts | 2009-09-03
http://dblp.l3s.de/d2r/resource/authors/Axel_Poller...

2 Axel Polleres - semantic... 3 facts | 2009-08-14
http://semanticweb.org/wiki/Axel_Polleres

3 Axel Polleres 23 facts | 2009-04-22
http://data.semanticweb.org/person/axel-polleres

4 Axel Polleres - wiki.sem... 3 facts | 2009-09-24
http://wiki.sembase.at/index.php/Axel_Polleres

5 DERI Galway - Dr. Axel P... 3 facts | 2009-06-09
http://www.deri.ie/about/team/member/axel_polleres/

6 Axel Polleres 33 facts | 2009-06-25
http://semanticweb.org/id/Axel_Polleres

7 Axel Polleres - wiki.sem... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Axel_Pollere...

8 Axel Polleres - wiki.sem... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Axel_Pollere...

9 Axel Polleres - wiki.sem... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Axel_Pollere...

10 Talk:Axel Polleres - wik... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Talk:Axel_Po...

11 Axel Polleres - wiki.sem... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Axel_Pollere...

12 Axel Polleres - wiki.sem... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Axel_Pollere...

13 Axel Polleres - wiki.sem... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Axel_Pollere...

14 Untitled document 9 facts | 2008-11-01
http://myopenlink.net/dataspace/kidehen/socialnetwo...

15 Axel Polleres - wiki.sem... 3 facts | 2009-09-03
http://wiki.sembase.at/index.php?title=Axel_Pollere...

16 Untitled document 6 facts | 2009-08-14
BOSS http://www.facebook.com/people/Axel-Polle...

17 Axel Polleres: books by ... 11 facts | 2009-09-07
BOSS http://www.bookfinder.com/author/axel-pol...

18 JeromeDL - Browse e-Libr... 1 facts | 2009-04-23
BOSS http://library.deri.ie/preview?author=mai...

19 Journal of Web Semantics... 1 facts | 2009-04-23
BOSS http://journalofwebsemantics.blogspot.com...

<- 1 2 ->

http://example.loc/document.rdf

Figure 1: Sig.ma screenshot for the query "Axel Polleres" when expanded to 30 sources – space usage is optimized using the property display configuration and reordering capabilities of the web interface

If the number of highly-scoring resource descriptions is low at this point, then an attempt is made to discover additional sources, based on the RDF data we have already retrieved and established to likely describe the target entity. We obtain new resource identifiers for the target entity using four methods: The URIs at the center of the selected resource descriptions are considered. If the resource descriptions include any `owl:sameAs` links, then the target URIs are considered. If the resource descriptions includes OWL inverse functional properties (IFPs) from a hardcoded list (e.g., `foaf:mbox` and `foaf:homepage`), then a Sindice index search for other resources having the same IFP value is performed. Finally we also employ the OKKAM service. OKKAM is an experimental service which assigns names to entities on the web [1]. OKKAM returns resource identifiers along with a confidence value. Any resource identifier discovered using these methods will be added into the *Query plan*, which will be then examined in the *refinement* step.

3.4 Consolidation

All selected resource descriptions are merged into a single entity profile by combining all key-value pairs from all resource descriptions into a single description. A reference to the original source is kept for each value.

Often different properties (keys in the key-value pairs that describe the entity) express the same thing. The next step is to consolidate the potentially large list of properties into a simpler list that is more meaningful to the user. In RDF, properties are named with URIs; we consider only the last segment ("local part") of the URI. By convention, this local part is usually a good name for the property, written in CamelCase or with underscores or dashes, which are converted back into a more readable string consisting of space-separated words. Next, we apply some well known English-language normalization heuristics on the property names.

Next, we apply a manually-compiled list of approximately 50 preferred terms. For example, we replace all of the following property names with the preferred term "web page": *work info homepage*, *workplace homepage*, *page*, *school homepage*, *weblog*, *website*, *public home page*, *url*, *web*. Special attention has been given to terms that can be used in customized ways in the user interface: labels, depictions (images), short descriptions, web links. Next, we drop a number of properties that are of little value in an end-user interface, e.g. `foaf:mbox_sha1sum` or `rdfs:seeAlso`.

After consolidation, properties are ranked. We use a simple ranking metric: the number of sources that have values for the property. This will push generic properties such as "label" and "type" to the top. The number of distinct values for the property is also factored in: properties where many sources agree on one or a few values (as observable with a person's name or homepage) receive a boost.

3.4.1 Value labelling and consolidation

For key-value pairs where the value is not a literal value, but a reference to another resource, a best-effort attempt is made to retrieve a good label for the resource: The original source RDF graph in which the resource was found is examined for typical label properties, such as `foaf:name`, `dc:title` or `rdfs:label`. If nothing is found, and it is a

URI, it will be resolved against the cache or the web, as described above. If nothing is found, and it is a URI, then the last part of the URI will be used in a manner similar as described above for property names.

This is an expensive process, as a typical entity profile can refer to dozens or hundreds of other entities, yet it is important for a good user experience. The labels also feed into further Sig.ma requests: when a user wants to follow a link to another entity, then the underlying resource identifier(s) as well as the label are used to submit a new Sig.ma request in order to produce the linked entity's profile. To achieve responsiveness despite the large required number of requests, labels are displayed incrementally using AJAX requests.

Property values with identical or very similar labels are collapsed into one value to improve the visual presentation. For example, several sources that describe a scientist can state that they have authored a certain paper, using different identifiers for the paper. Without label-based consolidation, the paper would appear several times because the identifiers are not the same. After label consolidation, it appears only once. Both identifiers are retained internally. A click on the paper link will cause a new Sig.ma search that has the label and both of the URIs as input. Since labels are retrieved and displayed incrementally, the value consolidation has to be performed in the same fashion.

3.5 Interactive source list refinement

After the entity profile is presented to the user, they can refine the result by adding or removing sources. Almost any entity profile initially includes some poor sources that add noise to the results. Mixed into the desired entity profile are other entities that have the same or a similar name, or that for other reason ranked highly in the text search portions. The user interface allows quick removal of these. Widgets for source removal exist in the list of sources, and next to each value that is displayed in the profile. If the profile shows a poor label or unrelated depiction for the entity, a quick click will remove the offending source, and the next-best label or depiction will automatically take its place if present. Other interactive activities include selecting a favorite label as well as reorganizing and removing property sections alltogether.

4. SIG.MA IMPLEMENTATION AND PERFORMANCE

The Sig.ma processing workflow is implemented in two layers, a Java backend, wrapped in a web application hosted in a Tomcat application server and a full MVC stack built in JavaScript. The backend exposes a RESTful API, which is used by the JavaScript layer through AJAX calls. It also represents a facade for the caching systems (memcached, HBase) and for other minor services. Averaged performance tests show that Sig.ma API takes around one second per 10 processed sources when serving RDF output, thus skipping the page rendering overhead. Most notably, Sig.ma seems to perform linearly with the number of sources, averaging a response time of 11 seconds per 100 processed sources.

5. RELATED WORK

Semantic Web search engines, such as SWSE [5], Swoogle [4], Falcons [2] or Sindice [7], are based on the common search paradigm, i.e., for a given keyword query (or more advanced queries) the goal is to return a list of ranked resources based on their relevance. Sig.ma, which is a search application built on top of Sindice, is positioned in another area more closely related to the "Aggregated Search" paradigm, since it provides an aggregated view of the relevant resources given a query [6]. One approach to aggregated search is to use different vertical searches (images, video, news, etc.) as input and to present the results into a single page e.g. as in Google Universal Search[6] or Yahoo! alpha[7]. [8] proposes to return "digest pages" which are virtual documents built from clustering and summarisation of the documents returned by a search engine. In contrast, Sig.ma propose to aggregate heterogeneous data gathered on the Web of Data into a single entity profile using Semantic Web data consolidation techniques. The user can then visualize the entity profile, but also enrich it with additional data sources and reuse it in other Semantic Web applications.

6. CONCLUSIONS

While Sig.ma is not the first data aggregator for the Semantic Web, its contribution is to show that exciting possibilities lie in a holistic approach for data discovery and consolidation. In Sig.ma, large scale semantic web indexing, logic reasoning, data aggregation heuristics, ad hoc ontology consolidation and, last but not least, user interaction and refinement, play together to provide entity descriptions which overcome many of the shortcomings of the current web of data.

7. REFERENCES

[1] B. Bazzanella, H. Stoermer, and P. Bouquet. An entity name system (ENS) for the semantic web. In *Proceedings of the European Semantic Web Conference*, 2008.

[2] G. Cheng and Y. Qu. Searching linked objects with Falcons: Approach, implementation and evaluation. *Int. J. Semantic Web Inf. Syst.*, 5(3):49–70, 2009.

[3] R. Cyganiak, M. Catasta, and G. Tummarello. Towards ECSSE: live Web of Data search and integration. In *Proceedings of the Semantic Search 2009 Workshop*, 2009.

[4] T. W. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng. Swoogle: Searching for knowledge on the semantic web. In *AAAI*, pages 1682–1683, 2005.

[5] A. Harth, A. Hogan, R. Delbru, J. Umbrich, S. O'Riain, and S. Decker. SWSE: Answers before links! In *Semantic Web Challenge, ISWC*, 2007.

[6] V. Murdock and M. Lalmas. Workshop on aggregated search. *SIGIR Forum*, 42(2):80–83, 2008.

[7] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *Int. J. of Metadata and Semantics and Ontologies*, 3:37–52, Nov. 10 2008.

[8] S. Sushmita and M. Lalmas. Using digest pages to increase user result space: Preliminary designs. In *SIGIR 2008 Workshop on Aggregated Search, Singapore*, 2008.

[6]`http://www.google.com/intl/en/press/pressrel/universalsearch_20070516.html`

[7]`http://au.alpha.yahoo.com/`