# An Entity Name System for Linking Semantic Web Data

Paolo Bouquet
DISI – University of Trento
Via Sommarive, 14
38100 Trento (Italy)
bouquet@disi.unitn.it

Heiko Stoermer DISI – University of Trento Via Sommarive, 14 38100 Trento (Italy) stoermer@disi.unitn.it

Daniele Cordioli ExpertSystem s.p.a. Via Virgilio, 56 41100 Modena (Italy) dcordioli@expertsystem.it

Giovanni Tummarello
DERI – National University of
Ireland
Lower Dangan
Galway (Ireland)
giovanni.tummarello@deri.org

#### **ABSTRACT**

In this paper, we argue that realizing the vision of the Semantic Web as an open and global decentralized knowledge space would greatly benefit from the availability of a large-scale service which may enable the systematic reuse of URIs across independently produced Semantic Web contents. Such a service, which we call an *Entity Name System* (ENS), might play for the Semantic Web the role that the DNS played for interlinking hypertexts on the Web. The key idea behing the ENS is that any tool/application for creating content (not only Semantic Web) may be empowered through simple plugins or extensions to interact with a network of ENS servers. Here we show how this has been done in a few prototypes for MS Word, Protégé and an editor for FOAF profiles.

# **Categories and Subject Descriptors**

H.4 [Information Systems Applications]: Miscellaneous

#### **General Terms**

Design

# **Keywords**

Semantic Web, Entity Name System, Identity

#### 1. INTRODUCTION

In a note from 1998, Tim Berners-Lee described the grand vision of the Semantic Web as follows:

Knowledge representation is a field which is currently seems to have the reputation of being initially interesting, but which did not seem to shake

the world to the extent that some of its proponents hoped. It made sense but was of limited use on a small scale, but never made it to the large scale. This is exactly the state which the hypertext field was in before the Web [...]. The Semantic Web is what we will get if we perform the same globalization process to Knowledge Representation that the Web initially did to Hypertext.

We understand this parallel as follows. Like the WWW provided a global space for the seamless integration of small hypertexts (or local "webs of documents") into a global, open, decentralized and scalable publication space; so the Semantic Web should provide a global space for the seamless integration of small knowledge bases (or local "semantic webs") into a global, open, decentralized and scalable knowledge space.

Today, as a result of many independent research efforts and commercial initiatives, relatively large and important knowledge repositories (like DBpedia.org, GeoNames, DBLP) have been made available which actually are (or can be easily tranformed into) local "semantic webs", namely graphs of resources connected through properties which are defined in some schema or vocabulary. Some of these repositories have already been interlinked through the Linked Data initiative (see linkeddata.org). However, it is difficult to claim that the interlinking of Semantic Web data is proceeding as fast as expected in a fully decentralized and open process. Why?

The argument we put forward in this paper is the following. On the one hand, the interlinking of local "webs of documents" into the WWW was largely made possible by a key enabling factor: the introduction of a global reference and addressing mechanism for locating and retrieving resources. This mechanism heavily depends on the availability of a global service, the DNS, which can resolve resource identifiers (URLs) into physical locations on the Internet. This is how one can be sure that, for example, a href link to a Web resource will be always resolved to the appropriate location on the Internet. On the other hand, the interlinking of local "semantic webs" is based on (i) the introduction of uniform identifiers for anything which can be named, includ-

ing concrete entities (people, geographical locations, events, artifacts, etc.) and abstract objects (concepts, relations, ontologies, etc.); and (ii) on the introduction of a generalized notion of link from simple hypertext links to any binary relation between resources. Whenever a statement is made about an entity  $e_1$  in any location of the network, then such a statement is in principle connected with any other statement made about the same entity elsewhere and independently, provided that (a) the same identifier (URI) is consistently used for it; or (b) an explicit mapping between the different URIs referring to the same entity is available. However, as to the former, to date no scalable and open service is available to make possible and to support a consistent reuse of identifiers for entities; as to the latter option, it seems to us that it cannot easily scale to a scenario in which billions of entities are identified through thousands of different URIs, as it would require a huge number of owl:SameAs statements and their use would be computationally too expensive. This situation undermines the practical possibility of a seamless interlinking of local knowledge into the global knowledge space envisage in the initial quote.

In this paper, we will try to defend the view that the practical realization of the grand vision of the Semantic Web as a huge graph of interlinked data would be much easier and faster if we could count on a service which, by analogy with the DNS, we call an Entity Name System (ENS), namely a service which stores and makes available for reuse URIs for any type of entity in a fully decentralized and open knowledge publication space. In this paper, we first describe how such a ENS should work; then we discuss the main issues and challenges associated with the design and implementation of a scalable and sustainable ENS; we present some preliminary data on the advantages of this approach for information integration on the Semantic Web, and briefly sketch some simple example of applications which are enabled to interact with the ENS for creating new content ready for being interlinked with pre-existing content.

# 2. AN ORDINARY DAY ON THE SEMANTIC WEB

Imagine an ordinary day on the Semantic Web:

- the University of Trento exports in RDF its bibliographic database;
- the WWW2008 conference organizers make available the metadata about authors and participants as part of the Semantic Web Conference (SWC) initiative;
- participants at WWW2008 upload their pictures on http://www.flickr.com/, and tag them;
- some participants at WWW2008 attend a talk on FOAF and decide to create their FOAF profile at http:// www.ldodds.com/foaf/foaf-a-matic and publish it on their web server;

• ...

At the end of the day, a lot of related material has been created. In principle, the newly created RDF content should allow Semantic Web programs to answer questions like: "Find

me which of my friends is attending WWW2008", "Find me pictures of Mike's friends who are attending WWW2008", "Find me the papers published by people of the University of Trento (or their friends) accepted at WWW2008", and so on. But, unfortunately, this can't be done. And the reason is that any time an entity (Mike, WWW2008, University of Trento, ...) is mentioned in one of the data sets, they are referred to through a different URI. And this does not allow browsing multiple RDF graphs based on the fact that the same resource is referred to by the same URI.

This scenario is quite typical of how Semantic Web content is produced today. Nearly like hypertexts in the pre-WWW era, every tool for creating semantic content mints new URIs for any resource, and this makes really difficult the bootstrap of the global knowledge space envisaged by Tim Berners-Lee. More and more attention is payed in reusing existing vocabularies or ontologies, but statements about specific resources (instances, individuals) cannot be automatically integrated, as there is nothing practically supporting the (desirable) practice of using a single global URIs for every resource, and resuing it whenever a new statement about it is made through some content creation application.

The Linked Data initiative is a very relevant and promising attempt to make this vision real. However, we believe that we need something more structural to support the process of interlinking RDF data. The ENS we will discuss in the next section is our proposed approach and solution for addressing this issue in a systematic and scalable way.

# 3. ENS: ARCHITECTURE

Our current prototype implementation of an ENS service is called Okkam and represents one node in a federated architecture, which is depicted as a cloud in the center of Fig. 1. The aim of the Okkam prototype is to provide a basic set of ENS functionality, i.e. searching for entities, adding new entities and creating new identifiers. The identifiers that Okkam issues are absolute URIs in the sense of RFC3986 [2], which makes them viable global identifiers for use in all current (Semantic) Web data sources; they furthermore are valid UUIDs, i.e. identifiers that guarantee uniqueness across space and time<sup>1</sup>, which prevents accidental generation of duplicates and thus also enables their use as primary keys e.g. in relational data sources.

What is illustrated in Fig. 1, and currently implemented as a single node, is planned to become a distributed system that is fully in line with the distributed nature of the (Semantic) Web. It is important to note that what we are propagating is an *entity-centric* approach, not an ENS-centric approach; this means, for example, that data sources which have issued their entities with OKKAM identifiers will continue to be integratable on the entity level, disregarding the existance or availability of an ENS server.

#### 3.1 Interaction with OKKAM

A critical feature of an ENS is to provide means for searching for the identifier of an entity. This step is strictly connected to the algorithm that supports the population of the sys-

<sup>&</sup>lt;sup>1</sup>See http://java.sun.com/j2se/1.5.0/docs/api/java/util/UUID.html for details

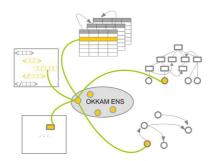


Figure 1: The ENS providing entity identifiers across system boundaries.

tem's repository with new entities. Indeed, when a query is submitted to the system, it has to decide if the query corresponds to an entity already stored (and return the information about it) or if a new entity has to be generated.

Figure 2 illustrates the standard use-case for the okkamization<sup>2</sup> of content, namely to query OKKAM for the existance of the entity at hand, and the re-use of a global identifier for this entity. This would usually be achieved through functionality provided by a client application, such as FOAF-O-MATIC or OKKAM4P (see Sect. 6) which accesses the OKKAM API, and presents (if available) a list of top candidates which match the description for the entity provided within the client application. If the entity is among these candidates, the client agent (human or software) uses the associated OKKAM identifier in the respective information object(s) instead of a local identifier. If the entity cannot be found, the client application can create a new entry for this entity in OKKAM and thus cause an identifier for the entity to be issued and used as described before.

# 3.2 Matching and Ranking in OKKAM

In the areas of databases and information integration, there is a substantial amount of related work that deals with the problem of detecting whether two records are the same or describe the same object.

The problem that we are dealing with in Okkam is very similar to these, because we see the process of searching for an entity as a matching problem of an entity description  $\Delta$  against the set EP of all entity profiles<sup>3</sup>.

The matching problem in OKKAM is however substantially different from the following points of view:

1. The description  $\Delta$  of the entity that is searched for can be generated by client applications that are of very different nature, i.e. they can have limited capabilities and e.g. only provide a simple query string, while others have additional background knowledge available or can provide (semi-) structured descriptions. It is thus not foreseeable which name/value pairs a  $\Delta$  contains.

- 2. Similiar to the first point, the set of entity profiles EP is untyped, semi-structured and may as well hold arbitrary values. The combination of (1) and (2) make a required solution very different from most record-linkage approaches in that these rely on fixed (and/or identical) schemas, because OKKAM cannot provide any meaningful schema.
- 3. The objective is not deduplication (or Merge/Purge etc.) but rather the production of a ranked list of candidate matches within a time frame of a few seconds. For this reason, unoptimized approaches that perform deduplication by iterating over EP in a serial fashion have to be avoided.

The setting of our problem is thus very similiar to what Pantel et al. describe about their *Guspin* system [12]: due to the high level of heterogeneity on the schema level (or in our case, the absence of such a level), we will pursue a purely data-driven approach for entity matching.

For our first prototype of OKKAM, we have implemented an exemplary matching and ranking algorithm. The objective was to provide an approach that solves the matching problem described above, and can serve as a baseline and benchmark for future developments.

Matching and ranking in OKKAM is a two-step process: first, a set of candidate matches is retrieved from the storage backend, which, in the second step, is ranked with respect to the input query. With this approach we try to alleviate the problem that while storage backends such as relational databases perform extremely well in its main purpose, the production of ranked query results is not a "native" feature and thus hard to achieve. Furthermore, it allows us to apply methods for ranking that such storage backends simply do not provide.

Due to the differences between the matching problem in OKKAM and much of the related work, we decided to pursue an approach that is both schema-independent (entities in OKKAM are not described with a fixed schema) and type-independent (entities are untyped). The solution we came up with is to see the EntityDescription  $\Delta_e$  of an entity as a type of document which we can compare against the EntityDescription  $\Delta_i$  that was provided in the input query. By computing a similarity between the two, and doing so for all candidate matches, we are able to provide a ranked query result.

<sup>&</sup>lt;sup>2</sup>We call *okkamization* the process of assigning an OKKAM identifier to an entity that is being annotated in any kind of content, such as an OWL/RDF ontology, an XML file, or a database, to make the entity globally identifiable.

<sup>&</sup>lt;sup>3</sup>Note that  $\Delta$  and EP are "compatible" for matching in the sense that every element  $E \in EP$  contains a  $\Delta$  by definition.

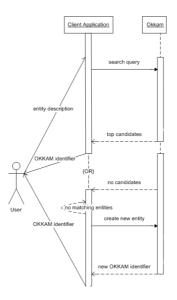


Figure 2: Sequence diagram of a standard interaction with Okkam.

The resulting algorithm, called StringSimilarityRank, is the following (with  $\Delta_e$  being denoted by de and  $\Delta_i$  by di):

```
d = concatenate(valuesOf(di))
forall candidates
   c = concatenate(valuesOf(de))
   s = computeSimilarity(d,c)
   rankedResult.store(s)
rankedResult.sort()
```

pairs that form part of  $\Delta$ , while concatenate() creates a single string from a set of strings; the combination of the two creates a "document" that can be matched against another, which is performed by the function computeSimilarity().

To compute the similarity between two descriptions, we have selected the Monge-Elkan algorithm [10] as the result of extensive testing and evaluation of different algorithms. The matching results that can be achieved with this approach are satisfactory as a baseline<sup>4</sup>.

It is important to note that this matching approach is completely generic and "un-semantic", in that it neither uses background knowledge nor any kind of type-specific heuristics to perform the described matching. This is in strong contrast with other approaches for matching that are currently pursued in the Linked Data community, which heavily rely on different kinds knowledge to perform such a match, and as a consequence, require a special heuristic to be developed for different schemas or entity types. One example is the matching of FOAF profiles based on the inverse functional property of the email hash, which is a highly specialized combination of knowledge about the entity type, its schematic representation, and the available data. While we believe that in the mid-term a well-designed set of specialized algorithms embedded in an adaptive system is a very promising approach, for the current prototype we explicitly pursued the goal of implementing an algorithm that is completely independent of any such knowledge, and thus can be used to any type of entity, in any representation.

#### ENS AND LINKED DATA

Currently, the most influencial and interesting approach for addressing the issues discussed so far in this paper is of course the Linked Data initiative<sup>5</sup>. Therefore, it is cru-The function valuesOf() returns the value parts of the name/valuecial that we discuss the similarities, the differences and the potential integration of our approach with the principles and practice of the Linked Data community.

> The two approaches are based on a largely overlapping common ground. Both the Linked Data initiative and our approach have their roots in the fact that to achieve the vision of the Semantic Web as a huge, distributed knowledge base, it is necessary to find a way to connect sets of statements about the same object stored in different (RDF/OWL) data sources, and that there currently exists a proliferation of identifiers for the same object which makes some kind of approach necessary to establish these connections, because a mere RDF graph merging does not produce the desired

> In [1], Tim Berners-Lee defines four major principles of linking data on the (Semantic) Web. These are

- 1. Use URIs as names for things.
- 2. Use HTTP URIs so that people can look up those names.
- 3. When someone looks up a URI, provide useful information.

<sup>&</sup>lt;sup>4</sup>Experiments and results are currently submitted for publication and under review, for the reason of which we do not report them in this article.

 $<sup>^5</sup>$ http://linkeddata.org/

Include links to other URIs. so that they can discover more things.

These principles represent – on a very high level – important commonalities between the two approaches. In fact, when analyzed, it becomes evident that the ENS approach (and Okkam as an implementation) is fully in line with these principles, because (1) URIs as identifiers (or proxies) for things are central to its design; (2) Okkam provides resolution of the URIs over HTTP; (3) when dereferenced over HTTP, an Okkam URI provides what we call an *Entity Profile*, i.e. the information that Okkam stores about an entity for making it findable and distinguishable from other entities; and (4), because at the center of Okkam there is the *Reference Repository*, which stores links to external data sources that contain data, information or knowledge *about* the entity, as a means to enable an entity-centric approach to accessing information on the (Semantic) Web.

It is important to note that we do not talk about generic "resources" as in pure RDF, but we distinguish "entities" and abstract objects, very much as traditional Description Logics or OWL [14] distinguish individuals and classes. The reason for this choice is that, even though the re-use of classes defined in OWL ontologies has not seen very large success, nevertheless these classes are already uniquely defined and dereferenceable through their fully qualified names, unlike individuals or – in our nomenclature – entities.

Having said this, we want of propose a critical look at the approach advocated by the Linked Data initiative. One very important difference with the use of an ENS is the fact that this initiative is working heavily on an ex-post approach. One of the core principles is to establish links between resources that are considered "identical" by means of an owl:sameAs statement between the different URIs denoting the entities. This has two important consequences. First of all, it must be noted that following such an approach in a highly dynamic and extremely rapidly growing environment such as the Web is truly a Sisyphean labor, because it requires the permanent search, analysis and alignment of new data, a task which – in theory – could partly be performed by the individual publishers of data but – in reality – is currently performed by the initiative itself, which introduces a form of centralization which is probably not intended.

Nevertheless, one must admit that the initiative has made a huge effort to link a very large amount of data. This result depends on specialized, data source-dependent heuristics to establish the owl:sameAs statements between resources, and it requires the statements to be *stored* somewhere (again, currently in a rather centralized fashion). These owl:sameAs statements, being introduced *ex-post*, have strong epistemic issues (how can one know that a certain entity is the same as another one, described somewhere else on the Semantic Web?) and also very strong logical consequences. As the OWL reference puts it: "Such an owl:sameAs statement indicates that two URI references actually refer to the same

thing: the individuals have the same 'identity' "7. In fact it appears that the current use of owl:sameAs in the Linked Data community is not entirely in line with this definition and uses owl:sameAs more like a Semantic Web substitute forx a hyperlink instead of realizing the full logical consequences.

Additionally, reasoning over owl:sameAs relations in distributed ontologies may be a complex task: the creation of owl:sameAs statements among the URIs recognized to identify "the same" entity implies the computation of the transitive closure, potentially across the whole of the Semantic Web. The transitive closure computation is known to belong to the NL computational complexity class [13, 15], an operation which may become overwhelming from a computational point of view at the moment when the number of created URIs and related owl:sameAs statements reach the limits of current DL reasoners. One approach to deal with this issue is to employ a data structure for representing equivalence classes, such as union-find<sup>8</sup>. However, it is obvious that such a structure (i) has to be built, with the problems mentioned before, and (ii) stored, which converts a system that applies such a technique into a centralized knowledge base that stores copies of distributed data. Another possible counter-argument to our criticism might be that this is all a merely technical problem and that with time the computational power will be sufficiently improved. However, in the moment the Semantic Web will continue growing as desired, a proliferation of URIs in combination with the computational effort of a transitive closure computation over owl:sameAs statements will in our opinion remain extremely costly and problematic, to say the least. Indeed, if the Semantic Web is going to approximate the present version of the web, it becomes hard to imagine a reliable system answering queries over massive, distributed transitive closures.

It is evident that the <code>owl:sameAs</code> approach does not address the problem of multiple identifiers for the same entity, on the contrary: it supports their proliferation. This leads us to the conclusion that the Linked Data approach as it is today may be more suitable for <code>browsing</code> than for reasoning or querying, and does thus is perhaps not the best way for realizing the vision of the Semantic Web as a large, distributed knowledge base.

The ENS, on the other hand, is clearly an a-priory approach. The goal is to work against a proliferation of identifiers for the same object in an architectural way. The benefits of this approach have previously been described, but the most striking one for the Semantic Web community is the fact that it will make the RDF graph merging what it was designed for: seamless integration of RDF data as the base of query answering over distributed data sets.

In the past, there have been critical opinions about the ENS approach which we would like to discuss at this point. Some of the issues mentioned are of social nature, for example that people would like to influence the appearance of "their" URI, or in general, have some control over the URI. This position

<sup>6</sup>http://esw.w3.org/topic/TaskForces/
CommunityProjects/LinkingOpenData/
EquivalenceMining

<sup>&</sup>lt;sup>7</sup>http://www.w3.org/TR/owl-ref/#sameAs-def

<sup>&</sup>lt;sup>8</sup>See e.g. http://en.wikipedia.org/wiki/Disjoint-set\_data\_structure for an informal introduction.

implies that a URI encodes some kind of identity, which is not straightforward and desirable in all cases: first of all, it mixes the identifier of an entity with knowledge about the entity, an issue that becomes evident when for example an email address is used as identifier and this email address changes; such an information-bearing identifier would become useless and lose its role as the sole and unique identifier for the object. Secondly, even though due to the current composition of Semantic Web data [7] discussions are often very concentrated about people data, of course the Semantic Web clearly goes beyond the domain of people. Control over their URI might be an issue for a selected group of users, but in a vast amount of other cases nobody would even consider argueing in this direction: today, IP addresses for internet hosts get assigned without anyone asking for e.g. the birth date of the owner being encoded in the address; phone numbers, serial numbers for products, license plates for cars, the address of a house or its geographical co-ordinates, and much more, are only a few further examples. In our opinion, an anonymous identifier, such as the ones issued by OKKAM, that is not bearing or encoding any information is much more suitable, helps prevent privacy issues, is less prone to becoming obsolete, and - again - in most cases is "socially" completely unproblematic because it is not people-centric.

A second, important criticism has been the appearance of the ENS as a "centralized" service that is in opposition with how things should be done in the Web. Our first response is that the ENS is not by accident named in analogy with the Domain Name System (DNS), a service that in the past has received criticism because of its centralized nature. However, to date, and in spite of all its issues, no viable alternative solution for the DNS mechanism has come into existance (or has been widely adopted). The semantics of the term "centralized" are also partly a matter of definition: if - as it is our goal for the evolution of Okkam - such a service is implemented in a distributed fashion by design, say, for example on top of a high-performance peer-to-peer infrastructure of Okkam nodes, then the argument of centralization begins to weaken. We do however think of an ENS as an important piece of backbone infrastructure with which the Semantic Web has the potential of reaching its original goals without being in the permanent race of trying to catch up with new data that have to be aligned ex-post.

This analysis might raise the impression that the two approaches, ENS and Linked Data, are two extreme positions that are mutually exclusive, even though they pursue the same high-level goals. We argue however that a co-existence of both approaches is possible. As long as the available amount of existing, unaligned (unlinked) data is still within manageable bounds, as we believe it is today but not for very much longer, the efforts of the Linked Data initiative can give important contribution precisely for aligning data that have been created so far. In the meanwhile, the ENS approach will be pursued and will - hopefully - catch up and be widely adopted due to the push that the European Commission's investment in the Okkam project represents for the next years, with the effect that in the mid-term the proliferation of identifiers for entities will decrease. These two efforts combined have the potential to converge to a status where the necessity for ex-post alignment of data will be required on an increasingly smaller scale (which is desirable because as we argued above it faces the enormous challenge of the growth of the Semantic Web in general), while "real" (URI-level) integration of Semantic Web data will drastically increase, and thus move the Semantic Web an important step forward, away from isolated knowledge base fragments towards that global knowledge space that it is envisioned to be.

# 5. ENS AND THE STRUCTURED DATA ON THE WEB

Given that the ENS needs to have metadata about entities to be able to perform entity matching, one could say as an extreme that the ENS should ideally contain every piece of structured information available on the Web. In practice, we foresee the ENS to use a different approach and specializing on its tasks by holding a restricted and specialized sets of attributes per entity while on the other hand leveraging existing infrastructures that provide awareness and near real time updates on information available on the Web of Data.

Sindice [11] is one such infrastructure which specializes in indexing both all the RDF documents made available on the Web, and structured information expressed using Microformats<sup>9</sup>. Sindice's task is to answer very low level queries such as "give me documents which contain descriptions of an entity named X" or "find documents which contain an entity having property Y equal to Z".

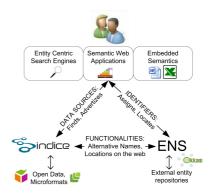


Figure 3: Sindice and ENS.

We envision the relationship between user applications, ENS and, for example, Sindice as depicted in Figure 3. Users are expected to operate without directed awareness either of the entity naming issues nor of the locations of semantically structured data on the web. Users are instead expected to operate directly on applications where semantics is used, e.g. as in Okkam4MSW or on foreseeable Entity Centric Search Engines.

The applications will then use the ENS only for two specific purposes: locating primary identifiers (or collections of identifiers) for entities, and to support the creation of new identifiers. Both when locating and when creating identifiers, just very specific metadata will be used, typically a standardized subset of the complete information that the application has about an entity. Thanks to the primary identifier, or the

 $<sup>^9\</sup>mathrm{See}$  http://microformats.org/.

list of identifiers, obtained by the ENS, the applications will then be able to effectively operate on the Semantic Indexing engines such as Sindice. The application will therefore be able to both locate the actual data sources, therefore obtaining the full metadata about the entities, and to advertise their newly created pieces of semantically structured data.

Clearly, it is envisioned that the ENS infrastructure and structured data indexing engines such as Sindice will themselves continuously interoperate to perform their respective tasks.

# 6. THREE ENS-ENABLED APPLICATIONS

To illustrate the viability and the usefulness of the approach, we have developed three exemplary prototypes that have been strategically selected from the area of content creation, and serve as means to achieve an *a priory* alignment of identifiers that we propagate in our approach. The reason for this selection was the fact that the success of ENS approach entirely depends on the availability of a critical mass of Okkamized content, and of course on the availability of tools for the creation of such content.

The three prototypes presented here are all available at http://www.okkam.org for download and test.

#### **6.1** OKKAM4P

The first tool is called Okkam4P [6], a plugin for the widelyused ontology editor Protégé. This plugin enables the creator of an ontology to issue individuals with identifiers from Okkam, instead of assigning local identifiers that bear the risk of non-uniqueness on a global scale. The choice for this tool was made based on two criteria, namely the target audience being rather 'expert' users of the Semantic Web, and, secondly, the very wide usage of the Protégé editor, which makes it a promising candidate for a rapid distribution of the tool.

The plugin essentially supports the assignment of a global unique identifier (the "Okkam ID") to a newly created individual, rather than relying on manual input of the user or the standard automatic mechanism of Protégé. To this end, it implements the use-case illustrated in Fig. 2: based on the data about an individual that are already provided in the KB developed by the user, it queries Okkam to see whether an identifier already exists which can be assigned to the new created individual, otherwise a new identifier would be created and used.

Access to the plugin is given through the context-menu of an individual, as depicted in Fig. 4. The plugin then guides the user through the search and selection process and finally replaces the local identifier for the entity with the one retrieved from the ENS. The result is an OWL ontology that is equipped with globally unique and re-usable identifiers and thus enables vastly simplified, automatic integrations with high precision. The plugin is available at the following URL: http://www.okkam.org/projects/okkam4p.

# **6.2** Foaf-O-Matic

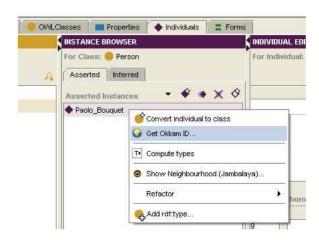


Figure 4: Assigning a global identifier to an individual.

The second application is called FOAF-O-MATIC [3], a WWW-based service for the creation of Okkamized FOAF<sup>10</sup> profiles. Indeed, FOAF is in our opinion one of the few real success stories of the Semantic Web so far, as it is one of the few applications that really contributed to the creation of a non-toy amount of RDF data, with the special restriction that the agreement on URIs for persons is extremely low [7]. As content creation tools for FOAF are mostly rather prototypical, we decided to create a completely new application that both serves the user due to state-of-the art technology and at the same time creates Okkamized FOAF profiles.

As we have explicated in [3], what is currently missing from FOAF is a reliable and pervasive way to identify "friends". The aim of creating the FOAF-O-MATIC application is not only to provide an alternative to the well-known foaf-a-matic application <sup>11</sup>. The focus of the new application is to allow users to integrate OKKAM identifiers within their FOAF document in a user-friendly way. In this way, it will be possible to merge more precisely a wider number of FOAF graphs describing a person's social networks, enhancing the integration of information and reach more easily the goal of the FOAF initiative.

A view of the application layout is given in Figure 5: it includes functions to re-use existing FOAF profiles (1), a form for describing oneself (2), the list of friends (3), and the form for adding friends (4) which initiates the ENS search process. The application is deployed and usable at the following URL: http://www.okkam.org/foaf-O-matic.

#### **6.3** OKKAM4MSW

The last ENS enabled application we present is a tool called OKKAM4MSW, a MS Word plugin for the globally unique identification of individuals in MS Word. Strictly speaking, this tool is not a Web (or Semantic Web) tool; however, it illustrates how the OKKAM concept may go beyond RDF/OWL content and support also the annotation of entities in non-structured formats/contents with global identifiers, this way (i) making possible a progress in tra-

<sup>10</sup>http://www.foaf-project.org

<sup>11</sup>http://www.ldodds.com/foaf/foaf-a-matic



Figure 5: Foaf-O-Matic The main interface of Foaf-O-Matic.

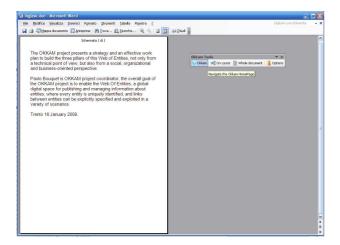


Figure 6: The Okkam4MSW interface.

ditional information retrieval with standard search engines, (ii) making easier the task of named entity recognition on information extraction and (iii) enabling the integration with RDF/OWL knowledge.

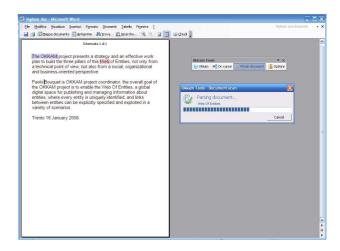


Figure 7: The search for recognized entities.

The great circulation of MS Word as a word processor is the main reason for the choice of implementing OkkaM4MSW. It allows for the unique identification of entities inside text documents created with MS Word. NLP and semantic technologies are used to detect the mention of an entity in the text and to extract contextual information that enables the matching decision within the OkkaM entity repository.

The plugin works as follows. The user writes a text using MS Word. The plugin identifies precisely or automatically entities by means of a thorough processing and linguistic disambiguation, which includes morphological, grammar, syntactic and semantic analyses of the text. All fundamental info for the disambiguation process, i.e. the whole system knowledge, is represented as a concept-based semantic network. Expert System's semantic network, called Sensigrafoő, is a rich conceptual representation of the language containing more than 400,000 concepts and millions of links between these concepts (for English only). All entity written with the first letter capitalized, are considered candidate entity and analysed within the structure of the sentence, in the context semantic and logical role. Through this process, the entity type identified are: people, companies, products, locations. The plugin can be accessed by means of a MS Word-integrated toolbar (Figure 6). Once the entity is identified, the plugin returns its unique identifier from the ENS - if found (Figure 7); otherwise a new identifier is generated. The analysis can be performed in real-time during document typing, or offline. The result is the option of uniquely choosing the entity which need be inserted into the text, or the a posteriori analysis of the whole document entities (Figure 7). The Okkam URIs are stored in the MS Word document, and becomes available for any use, including indexing from search engines.

#### 7. RELATED WORK

There are currently two major approaches which can be considered relevant for the topic described in this paper.

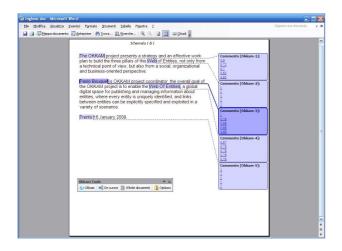


Figure 8: The outcome of the search for entities.

The first is, of course, the *Linking Open Data Initiative*, which was extensively discussed and compared with the ENS approach in Section 4.

The second is presented in Jaffri et al. [8]. In their work resulting from the ReSIST project, these authors recently came to a conclusion similar to the one we expressed in [4, 5], namely that the problem of proliferation of identifiers and the resulting coreference issues should be addressed on an infrastructural level. As a solution, they propose what they call a Consistent Reference Service. While we share this general view, their point about URI potentially changing "meaning" depending on the context in which they are used, is philosophically disputable: the fact that several entities might be named in the same way ("Spain" the football team, "Spain" the geographic location) must not lead to the conclusion that they can be considered the same unter certain circumstances<sup>12</sup>. Furthermore, their implementation of "coreference bundles" which establish identity between entitites, are in fact very similar to a collection of owl:sameAs statements, that we discussed below.

Recently, it has been proposed to use an Semantic Web Indexing Engine like Sindice to locate the best URI for a person<sup>13</sup>. The approach consists in lunching a query either on the full name or on the SHA1 sum of the email of the person that one seeks to find an identifier for. The documents which are returned by Sindice are then analyzed to locate FOAF profiles which have the foaf:primaryTopic having a matching name or email address. The URI of the primary Topic node is then used as identifier. The approach is certainly interesting, but it is currently restricted to the identification of persons which have a properly crafted FOAF file available online, furthermore assuming that there will be no conflicting or malicious information among the returned documents.

# 8. CHALLENGES AND CONCLUSIONS

In the paper, we presented the idea and the results of a test on ontology integration with a simple prototype of the ENS. However, designing, implementing and making available the ENS on a global scale involves some very difficult scientific and technological challenges. Here we list some chalenges, and discuss how we think to address them in the FP7 project OKKAM.

The first big challenge has to do with the idea of recognizing that an entity named in some content (e.g. in an RDF graph) is the same as an entity stored in the ENS repository. This entity matching problem, which is obviously related to well-known problems is several disciplines (e.g. named entity recognition, coreference, object consolidation, entity resolution), is one of the most important reasons why developing a working and scalable ENS is intrinsically more difficult than the DNS. The solution which we will explore in the project is based on two ideas:

- on the one hand, any entity will be stored in a ENS server with its URI and a non fixed set of attribute-value pairs. This "profile" of the entity is not meant to be a source of correct and up-to-date information about the entity, but as a minimal collection of information that a human or a program can use to identify an entity, or to distinguish between two entities. The matching algorithm we used for the test in this paper is quote simple, and is based on string similarity algorithms; still, the results are encouraging We plan to refine this method with heuristic rules, possibily specialized for broad categories of entities (e.g. people, organizations, locations, events, products, etc.);
- on the other hand, we will investigate how we can automatically build a contextual profile for entities named in content specified in different formats (e.g. text, HTML or XML files, RDF/OWL databases, relational databases) and how such a profile can be used for matching the entity against the profile available in an ENS server.

A second issue has to do with bootstrapping the service. This problem has two dimensions. First, we need to make sure that the ENS is pre-populated with a significant number of entities, so that there is a reasonable chance that people find a URI to reuse in their application; this will be done by implementing tools for importing entities (and their profiles) from existing sources (e.g. from DBpedia.org, DBLP, factbooks, GeoNames, etc.). Second, and even more important, we need to make sure that the interaction with the service is integrated in the largest possible number of common application for creating content. In Section 6 we described two simple examles of how we imagine this interaction should happen; however, it is our plan to extend the idea also to non Semantic Web tools, like office applications, web-based authoring environments (including forums, blogs, multimedia tagging portals, and so on). This approach should make the interaction with the ENS very easy (ideally, the use need not be involved, though he or she of course must be aware of it, see below for this issue) and will slowly introduce the good practice of OKKAMizing any new content which is created on the Web.

<sup>&</sup>lt;sup>12</sup>see e.g. Kripke [9]

<sup>13</sup> See http://www.wikier.org/blog/using-sindice-to-get-the-best-uri-for-a-person.

A third big issue has to do with the scalability of the proposed solution. Indeed, the number of entities which people might want to refer to on the Web is huge, and the number of requests that the ENS might receive from OKKAM enabled applications can be extremely high. For this reason, the architecture which we envisage for the ENS is totally distributed and decentralized, and is very similar to the DNS architecture. However, there is a problem here: while the DNS is organized hierarchically (mainly geographically), we don't see any obvious way to subdivide the domain of entity names in a hierarchical way. By design, we decided not to impose any conceptual schema on entities, as this would mix the naming service with a viewpoint on what is true about an entity. But we can't assume that the ENS architecture is completely flat, as for example it would be extremely hard to check whether an entity which someone is trying to create through a ENS server already exists somewhere else. This is perhaps the most important open issue in the proposed approach.

Last but not least, we want to mention two non-technical related issues. The first has to do with acceptance: how will we convince people to adopt the ENS? We already had some feedback on the concern that the ENS might become a way to track people or organizations, and therefore we need to make sure that the benefits outnumber the concerns by proving the advantages of the service in a few very visibel and popular domains. The secnd issue has indeed to do with the general problem of guaranteeing privacy and security of the ENS. As to this respect, it is important that we do not raise the impression that the ENS is about storing lots of information about anything. The profiles which we will store will be minimal, and will have the only function of supporting reasonably robust matching techniques. Also, we need to make sure that people have some degree of control on what can be stored in a profile, what cannot, and on what can be stored for improving matching but should never be returned as the result of a query to the ENS.

We are aware that the challenges are quite ambitious, but in our opinion the ENS has the potential to become the factor which will enable the creation of the global, open and decentralized knowledge space which Tim Berners-Lee envisions in the quote we reported at the beginning of the paper.

# 9. ACKNOWLEDGMENTS

This work is partially supported by the by the FP7 EU Large-scale Integrating Project **OKKAM** – **Enabling a Web of Entities** (contract no. ICT-215032). For more details, visit http://www.okkam.org.

# 10. REFERENCES

- T. Berners-Lee. Design Issues Linked Data.
   Published online, May 2007.
   http://www.w3.org/DesignIssues/LinkedData.html.
- [2] T. Berners-Lee, R. Fielding, and L. Masinter. RFC 3986: Uniform Resource Identifier (URI): Generic Syntax. IETF (Internet Engineering Task Force), 2005. http:
  - //www.gbiv.com/protocols/uri/rfc/rfc3986.html.
- [3] S. Bortoli, H. Stoermer, and P. Bouquet. Foaf-O-Matic Solving the Identity Problem in the FOAF Network.

- In Proceedings of the Fourth Italian Semantic Web Workshop (SWAP2007), Bari, Italy, Dec.18-20, 2007, December 2007.
- http://CEUR-WS.org/Vol-314/43.pdf.
- [4] P. Bouquet, H. Stoermer, and D. Giacomuzzi. OKKAM: Enabling a Web of Entities. In i3: Identity, Identifiers, Identification. Proceedings of the WWW2007 Workshop on Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007., CEUR Workshop Proceedings, ISSN 1613-0073, May 2007. online http://CEUR-WS.org/Vol-249/submission\_150.pdf.
- [5] P. Bouquet, H. Stoermer, M. Mancioppi, and D. Giacomuzzi. OkkaM: Towards a Solution to the "Identity Crisis" on the Semantic Web. In Proceedings of SWAP 2006, the 3rd Italian Semantic Web Workshop, Pisa, Italy, December 18-20, 2006. CEUR Workshop Proceedings, ISSN 1613-0073, online http://ceur-ws.org/Vol-201/33.pdf, December 2006.
- [6] P. Bouquet, H. Stoermer, and L. Xin. Okkam4P A Protégé Plugin for Supporting the Re-use of Globally Unique Identifiers for Individuals in OWL/RDF Knowledge Bases. In Proceedings of the Fourth Italian Semantic Web Workshop (SWAP2007), Bari, Italy, Dec. 18-20, 2007, December 2007. http://CEUR-WS.org/Vol-314/41.pdf.
- [7] A. Hogan, A. Harth, and S. Decker. Performing object consolidation on the semantic web data graph. In i3: Identity, Identifiers, Identification. Proceedings of the WWW2007 Workshop on Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007., 2007.
- [8] A. Jaffri, H. Glaser, and I. Millard. Uri identity management for semantic web data integration and linkage. In 3rd International Workshop On Scalable Semantic Web Knowledge Base Systems. Springer, 2007.
- [9] S. Kripke. Naming and Necessity. Basil Blackwell, Boston, 1980.
- [10] A. E. Monge and C. Elkan. An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In DMKD, pages 0-, 1997.
- [11] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: A document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3(1), 2008.
- [12] P. Pantel, A. Philpot, and E. H. Hovy. Matching and Integration across Heterogeneous Data Sources. In Proceedings of the 7th Annual International Conference on Digital Government Research, DG.O 2006, San Diego, California, USA, May 21-24, 2006, pages 438-439, 2006.
- [13] C. Papadimitriou. Computational Complexity. Addison-Wesley, 1994.
- [14] P. Patel-Schneider, P. Hayes, and I. Horrocks. Web Ontology Language (OWL) Abstract Syntax and Semantics. Technical report, W3C, February 2003. http://www.w3.org/TR/owl-semantics/.
- [15] M. Sipser. Introduction to the Theory of Computation. PWS Publishing, 1997.