# An Entity Name System (ENS) for the Semantic Web*

Paolo Bouquet[1], Heiko Stoermer[1], and Barbara Bazzanella[2]

[1] Dipartimento di Ingegneria e Scienza dell'Informazione – University of Trento
Via Sommarive, 14 – 38050 Trento, Italy
[2] Dipartimento di Scienze della Cognizione e della Formazione – University of Trento
Via Matteo del Ben, 5 38068 Rovereto (TN), Italy
bouquet@disi.unitn.it, stoermer@disi.unitn.it,
b.bazzanella@email.unitn.it

**Abstract.** In this paper, we argue that implementing the grand vision of the Semantic Web would greatly benefit from a service which can enable the reuse of globally unique URIs across semantic datasets produced in a fully decentralized and open environment. Such a service, which we call *Entity Name System* (ENS), stores pre–existing URIs and makes them available for reuse mainly – but not only – in Semantic Web contents and applications. The ENS will make the integration of semantic datasets much easier and faster, and will foster the development of a whole family of applications which will exploit the data level integration through global URIs for implementing smart semantic-based solutions.

## 1 Introduction

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

> Knowledge representation is a field which 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 [http://www.w3.org/DesignIssues/RDFnot.html].

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")

* 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://fp7.okkam.org. The authors are also very grateful to Claudia Niederee and Rodolfo Stecher for their support in distilling the ENS core concepts.

S. Bechhofer et al.(Eds.): ESWC 2008, LNCS 5021, pp. 258–272, 2008.

into a global, open, decentralized and scalable *web of documents*, so the Semantic Web should provide a global space for the seamless integration of semantic repositories (or "local semantic webs") into a global, open, decentralized and scalable *web of knowledge bases*.

Today, as a result of many independent research projects and commercial initiatives, relatively large and important knowledge repositories have been made available which actually are (or can be easily tranformed into) "local semantic webs", namely sets of statements connecting to each others any type of resource through properties which are defined in some schema or vocabulary. DBpedia, GeoNames, DBLP, MusicBrainz and the FOAF profiles are only a few examples of knowledge bases which have been made available in semantic web formats (RDF/OWL); but any social network, digital library metadata collection, commercial catalog and in principle any relational database could be easily (and mostly syntactically) transformed into a "local semantic web" by exposing its data on the Web in RDF/OWL. Apparently, the necessary building blcks of the Semantic Web are available. So why is the integration of these local "semantic webs" not progressing as expected?

The argument we propose in this paper is the following. The integration of local "webs of documents" into the WWW was largely made possible by a key enabling factor: the introduction of a global and unique addressing mechanism for referring to and locating/retrieving resources. This addressing space relies on the existence of a service like the Domain Name System (DNS[1]) which maps any Uniform Resource Locator (URL) into a physical location on the Internet. This is how we be sure that, for example, a document with a suitable URL will be always and unmistakably located and retrieved, and that a `href` link to that resource (through its URL) will always be resolved to the appropriate location (even when the physical location of the resource may have changed, e.g. it was moved to another machine with a different IP address). The integration of "local semantic webs" is based on a very powerful generalization of what can be addressed on the web: from information objects (like HTML pages, documents, servers, etc.) to any type of object, including concrete entities (like people, geographical locations, events, artifacts, etc.) and abstract objects (like concepts, relations, ontologies, etc.). This leads to a higher level concept of integration: if two (or more) independent "local semantic webs" make statements about the same entity $e_1$, then these statements should be connected to each other, this way combining the knowledge about $e_1$ provided separately in the two sources. For this seamless integration of "local semantic webs" to become possible, it is required that independent data sources address (i.e. refer to) the same resource through the same URI. But this is not the case today, and a new URI is minted for a resource every time it occurs in a RDF/OWL knowledge base.

As we will show, there are two general views on how to address this issue. The *ex post* view is based on the idea that the multiplicity of URIs for the same entity is not bad *per se*, and that an appropriate solution is the creation of identity statements between any URIs which have been created for the same entity; this

[1] See `http://www.ietf.org/rfc/rfc1034.txt`

approach is well exemplified by the Linked Data initiative[2]. The *ex ante* approach is based on the idea that the proliferation of URIs for the same entity should be limited from the outset, and that a suitable solution should support the widest possible use (and – most importantly – reuse) of globally unique URIs. Though the two solutions are in principle not mutually exclusive, in another paper [12] we presented some arguments for preferring the *ex ante* over the *ex post* view. In this paper we present a technical solution for supporting the *ex ante* view based on an *Entity Name System* (ENS) for the Semantic Web, an open and global service which can be used within existing applications to support the creators/editors of semantic web content to (re)use the same globally unique URI for referring to the same entity in a systematic way.

## 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 5th European Semantic Web Conference (ESWC2008) makes available the metadata about authors and participants as part of the Semantic Web Conference (SWC) initiative;
- participants at ESWC2008 upload their pictures on `http://www.flickr.com/` and tag them;
- some participants at ESWC2008 attend a talk on FOAF and decide to create their FOAF profiles at `http://www.ldodds.com/foaf/foaf-a-matic` and publish them on their web servers;
- ...

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 ESWC2008", "Find me pictures of Fausto's friends who attended ESWC2008", "Find me the papers published by people of the University of Trento (or their friends) accepted at ESWC2008", and so on. But, unfortunately, this can't be done. And the reason is that every time an entity (Fausto, ESWC2008, University of Trento, ...) is mentioned in one of the data sets, it is referred to through a different URI. And this does not allow RDF graph merging based on the fact that the same resource is referred to by the same URI. This is an instance of the so–called problem of identity and reference on the Semantic Web.

This scenario is quite typical of how Semantic Web content is produced today. Nearly like hypertexts in the pre-WWW era, any tool for creating semantic content mints new URIs for every resource, and this seriously hinders the bootstrapping of this global knowledge space called Semantic Web as it was originally envisioned. More and more attention is payed to reusing existing vocabularies or

[2] See `http://linkeddata.org/`

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 URI for every resource, and reusing it whenever a new statement about it is made through some content creation application.

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: A Prototytpe Architecture

Our current prototype implementation of an ENS service is called Okkam[3] and represents one node in a federated architecture, which is depicted as a cloud in the center of Figure 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 [1], 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[4], which prevents accidental generation of duplicates and thus also enables their use as primary keys, e.g. in relational data sources.

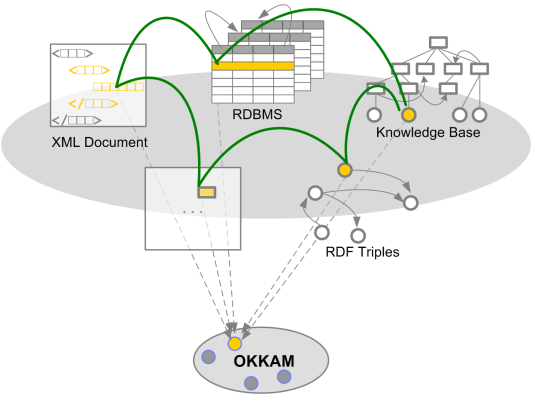


**Fig. 1.** The ENS providing entity identifiers across system boundaries

What is illustrated in Figure 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.

[3] As a variation of the **Ockham's razor**, we propose the **Okkam's razor** as a driving principle: "*entity identifiers should not be multiplied beyond necessity*" ...

[4] See `http://java.sun.com/j2se/1.5.0/docs/api/java/util/UUID.html` for details.

### 3.1 Interaction with Okkam

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

The standard use-case for the *okkamization*[5] of content goes as follows. A client application (such as FOAF-O-MATIC or OKKAM*4P*, presented in Section 5) 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

The problem of searching for an entity in an ENS can be viewed as the problem of matching an entity description $\Delta$ from an external source against the set $EP$ of all entity profiles stored in the OKKAM entity repository[6]. The setting of our problem is thus very similiar to what Pantel et al. describe about their *Guspin* system [2]: 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, we have implemented an exemplary matching and ranking algorithm, whose objective is to provide a first solution of 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

[5] 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.

[6] Note that $\Delta$ and $EP$ are "compatible" for matching in the sense that every element $E \in EP$ contains a $\Delta$ by definition.

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.

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()
```

The function `valuesOf()` returns the value parts of the name/value 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 [3] as the result of extensive testing and evaluation of different algorithms [4]. The matching results that can be achieved with this approach are satisfactory as a baseline, as will be evident from Sect. 4.

This matching approach is completely general and "a-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 e.g. 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.

## 4 An Experiment in ABox Integration

To illustrate one possible application of the OKKAM infrastructure, we performed an ontology integration experiment with the Semantic Web data which cover information about papers, schedules, attendees, etc. of the two recent Semantic Web conferences, namley ISWC2006 and ISWC2007[7].

While this is not the "typical" application of OKKAM, as it is an *ex-post* alignment which we do not propagate as best practice, we set up this experiment

[7] These datasets are available at `http://data.semanticweb.org`; for future reference, we made available a copy of the datasets at `http://okkam.dit.unitn.it/swonto/`

to test and improve the performance of the current OKKAM prototype and of the implemented methods for entity matching and for ranking results.

The aim of the experiment is to perform fully automated object consolidation on entities of type *foaf:Person*, to evaluate several aspects of this process, and consequently, to establish a threshold for entity identity on which processes such as automatic alignment can rely. In the following, we evaluate three steps:

1. Establishing threshold $t_{fp}$, which can be considered a "good" value below which a best match found by OKKAM should be considered a false positive.
2. Establishing a golden standard $g$ to evaluate the results of the merging process grounded on the threshold $t_{fp}$.
3. Performing an unsupervised ontology merge and analyzing the results.

### 4.1 Establishing an Identity Threshold

In OKKAM, deciding whether an entity $e$ matches a query $q$ relies on a similarity threshold $t_{fp}$ below which $e$ should be considered a false positive.

To fix this threshold, we ran the system on a set of example queries (all person-entities from the ISWC2006, ESWC2006 and ISWC2007 metadata sets). For each query, the system returns an OKKAM URI and a corresponding similarity value. Subsequently we checked manually the performance of the system, comparing the data available about the source URI with the Okkam URI to verify whether the match was correct or false.

Subsequently we evaluate how the performance of the system changes, varying the threshold on the range of similarity calsses ($t_1 = s_1, ..., t_j = s_j$) and for each class we compute the contingency table (see Table 1), including values for True Positive ($TP_j$), True Negative ($TN_j$), False Positive ($FP_j$) and False Negative ($FN_j$).

**Table 1.** Contingency table

| $S_j$ | Expert assigns YES | Expert assigns NO |
|---|---|---|
| System assigns YES | $TP_j$ | $FP_j$ |
| System assigns NO | $FN_j$ | $TN_j$ |

Here, $TP_j$ (True Positive with respect to the threshold $t_j$) is the number of entities correctly identified by the system when the threshold is $t_j$, $TN_j$ is the number of entities that the system correctly did not identify when the threshold is $t_j$, $FP_j$ is the number of the entities that have been incorrectly identified by the system when the threshold is $t_j$ and $FN_j$ is the number of entities that the system incorrectly did not identify.

The first evaluation that we performed was comparing the trend of TP with respect to FP. This analysis is motivated by the aim of our investigation to find the threshold that results in a minimum of FP but preserves a good level of TP. In general, if the number of FP is too high, the negative effects are two-fold: on

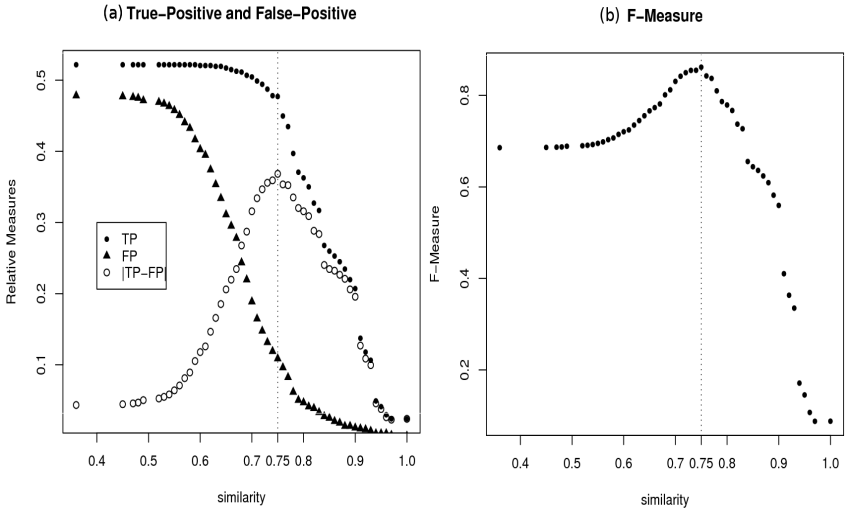


**Fig. 2.** Evaluation Measures

the one hand the results returned by the system would be polluted by irrilevant information, while on the other hand if the same threshold is used to perform the entity-merging, two different entities would be collapsed. The latter is a very undesirable circumstance because it leads to the loss of one of the two entities, and in the assignment of wrong information to the other (wrong merge/purge).

In order to determine an acceptable TP-FP trade-off we adopt a distance measure between TP and FP (the absolute value of the difference between TP and FP, $|TP - FP|$ or *Manhattan distance*) to establish the value of similarity in respect to this distance is maximized. In Figure 2(a) we plot TP and FP and the absolute value $|TP - FP|$. The graph shows that FP decrease more rapidly compared to TP when the similarity increses and the trend of difference $|TP - FP|$ shows a like normal distribution with a peak in correspondence to the maximum on a level of similarity equal to 0.75. On this level the system presents TP=0.47 and FP=0.10.

In order to confirm our result we evaluated the performance of the system measuring its effectiveness by means of Precision (P), Recall (R) and F-Measure (F)[8]. For each similarity class we calculate these evaluation measures to find which similarity value ensures the best performance of the system. We present the results relative to the F-Measure that give an overall description of the performance. In Figure 2(b) we show how the F-Measure varies as a function of similarity. We can see that the F-Measure increases up to a level of similarity equal to 0.75 and then dicreases rapidly. This evidence confirms the same result of the first analysis, indicating as the best threshold $t_{fp} = 0.75$. On this level we register a value of the F-Measure equal to 0.86, corresponding to P=0.81 and

[8] See `http://en.wikipedia.org/wiki/F-measure` for an overview of these performance measures from the field of Information Retrieval.

**Table 2.** Performance for t=$t_{fp} = 0.75$

| $t_{fp} = 0.75$ | TP | TN | FP | FN | P | R | FM |
|---|---|---|---|---|---|---|---|
| | 0.47 | 0.36 | 0.10 | 0.04 | 0.81 | 0.91 | 0.86 |

R=0.91. Table 2 summarizes the performance of the system when the threshold is $t_{fp} = 0.75$.

### 4.2 Evaluating the Ontology Merge

In order to evaluate the performance of the system with respect to the results of the merging process, we have to define a benchmark that we consider as the golden standard in our analysis.

For this purpose we took into account two (ISWC2006 and ISWC2007) out of three Semantic Web ontologies considered in the first phase of our evaluation analysis.

As a first step, we compare manually the two data sets to detect which URLs in the two datasets refer to the same real world entities (persons). This comparison returns the number ($g = 48$) of entities that the system should be able to consolidate, and represents the golden standard of our analysis.

In the second step of the analysis we perform an automatic merge of the same data sets (ISWC2006 and ISWC2007), and compare this merge to the golden standard. In Table 3 we report the results of our analysis respect to three exemplary thresholds which we examined.

If we consider the first data column in Table 3 in which we have the results respect to a value of $t_{fp} = 0.75$, we notice that the correct mappings amount to 46, which – compared to the golden standard of $g = 48$ – shows that the system returns almost all the correct mappings. However the number of false positives is still quite high and it reduces precision to $P = 0.65$. In other words, the system recognises some wrong mappings, which requires us to search for another (more safe) threshold that guarantees a lower number of FP, but preserving a satisfying number of TP. Table 3 shows that $t_{fp} = 0.90$ increases precision

**Table 3.** Results of the merging process

| | $t_{fp} = 0.75$ | $t_{fp} = 0.90$ | $t_{fp} = 0.91$ | Golden standard |
|---|---|---|---|---|
| Total Positives | 70 | 68 | 43 | 48 |
| True Positive | 46 | 45 | 25 | 48 |
| True Negative | 380 | 385 | 405 | 403 |
| False Positive | 24 | 20 | 20 | 0 |
| False Negative | 1 | 1 | 1 | 0 |
| Precision | 0.66 | 0.69 | 0.56 | 1 |
| Recall | 0.98 | 0.98 | 0.96 | 1 |
| F-Measure | 0.78 | 0.81 | 0.7 | 1 |

without sacrificing substantially TP, while $t_{fp} = 0.91$ leads to a degeneration of the results.

Summing up, our experiment showed that it is possible to move from two data sources that should set-theoretically present a certain overlap but syntactically do not[9], to a situation where good recall of matches can be reached through an alignment against OKKAM. The approach presented here requires no ad-hoc implementations or knowledge about the representation of the entities, as opposed to other approaches, such as [5] or the ones described in Section 6.

## 5 Two ENS-enabled Applications

To illustrate the viability and the usefulness of the approach, we have developed two exemplary applications that both have been strategically selected from the area of content creation, and serve – in contrast to the experiment described in Sect. 4 – 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 the ENS approach depends entirely on a certain saturation of suitable content ("critical mass"), and in effect on the availability of tools for the creation of such content.

### 5.1 Okkam*4P*

The first tool is called OKKAM*4P* [6], a plugin for the widely-used 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.

Based on the data about an individual provided in the KB developed by the user. The plugin queries OKKAM to see whether an identifier already exists which can be assigned to the newly created individual, otherwise a new identifier is created and returned.

Access to the plugin is given through the context menu of the individual, as depicted in Figure 3. 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`.

### 5.2 Foaf-O-Matic

The second application is called FOAF-O-MATIC [7], a WWW-based service for the creation of okkamized FOAF[10] profiles. Indeed, FOAF is in our opinion one

[9] In fact, the two data sources present an overlap of zero identifiers for person entities.
[10] `http://www.foaf-project.org`

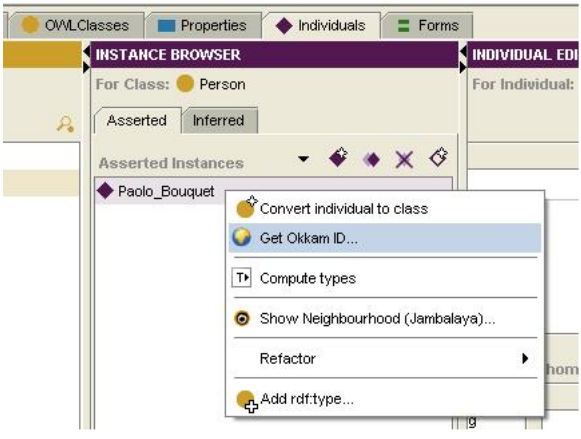


**Fig. 3.** Assigning a global identifier to an individual

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 [5]. As content creation tools for FOAF are mostly rather prototypical, we decided to create a completely new application that both serves the user with state-of-the-art technology and at the same time creates okkamized FOAF profiles.

As we have discussed in [7], what is currently missing from FOAF is a reliable and pervasive way to identify "friends". 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 advancing toward the goal of the FOAF initiative.

A view of the application layout is given in Figure 4: 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 Related Work

The problem 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 (the *entity matching* problem) is obviously related to well-known problems in several disciplines[11] (e.g. named entity recognition, coreference, object consolidation, entity

[11] See [4] for a more detailed discussion of related work.


**Fig. 4.** FOAF-O-MATIC The main interface of FOAF-O-MATIC

resolution). In the areas of database 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. However, the matching problem in OKKAM is different for the following reasons:

1. the description of the entity that is searched for can be generated by client applications that are of very different nature. Some (like a text editor) may only provide a simple query string, while others (like ontology editors) may provide (semi-) structured descriptions. It is thus not foreseeable which key/value pairs a description contains;
2. the set of entity profiles stored in OKKAM is untyped, semi-structured and may as well contain arbitrary values. This aspect, combined with the previous one, makes the ideal OKKAM solution very different from most record–linkage approaches, as they rely on fixed (and/or identical) schemas, whereas OKKAM does not;
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 fraction of second. For this reason, unoptimized approaches that perform deduplication by iterating over entity profiles in a serial fashion must be avoided.

As part of the solution, 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.

For dealing with the proliferation of indentifiers in the Semantic Web, there are currently at least two major approaches which are very relevant.

Jaffri et al. [8], in their work resulting from the ReSIST project, recently came to a conclusion not very different from the one to we advoceated in [9,10], namely that the problem of the proliferation of identifiers (and the resulting coreference issues) should be addressed on an infrastructural level; consequently they propose what they call a *Consistent Reference Service*. While we share this general view, their point about URIs 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* under certain circumstances[12]. Furthermore, their implementation of "coreference bundles" for establishing identity between entitites is in fact very similar to a collection of `owl:sameAs` statements (see below).

Another notable approach is the effort of the *Linking Open Data Initiative*[13], which has the goal to "connect related data that wasn't previously linked". The main approach pursued by the initiative is to establish `owl:sameAs` statements between resources in RDF. While the Linked Data community has made a huge effort to interlink a large number of datasets, our view is that this approach is not optimal to realize the vision of the Semantic Web as a large, decentralized knowledge base. First of all, a very large number of `owl:sameAs` statements is necessary, and it grows with the number of different URIs which are available for the same entity; second, querying distributed datasets cannot be done by simple SPARQL queries, as it must be combined with some form of reasoning (unless all the implied identity statements are computed beforehand and stored with the data); third, it sounds quite unrealistic that users will spend time in creating identity statements about their local data. However, we would like to stress that the ENS approach and the Linked Data initiative are not at all mutually exclusive, as OKKAM identifiers can be easily linked to other non-okkamized datasets through `owl:sameAs` statements, and `owl:sameAs` statements can be used in OKKAM to generate aliases for an OKKAM identifier. See [12] for a more thoroughly discussion of the relationship between the two approaches.

## 7 Challenges and Conclusions

In the paper, we presented the idea and the results of a test on ontology integration with a 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 challenges, and discuss how we plan to address them in the FP7 project OKKAM.

[12] See e.g. Kripke [11].

[13] See `http://linkeddata.org/`

In the previous section we alrady discussed the *entity matching* problem. 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 will find a URI to reuse in their applications; this will be done by implementing tools for importing entities (and their profiles) from existing sources. 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 applications for creating content. In Section 5, we described two simple examples 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 or web-based authoring environments (including forums, blogs, multimedia tagging portals, and so on). This approach should make interaction with the ENS very easy, sometimes even transparent, and will slowly introduce the good practice of OKKAMizing any new content which is created on the Web.

A third big issue has to do with 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 be exposed to can be extremely high. For this reason, the architecture we envisage for the ENS is distributed and decentralized.

Last but not least, there are two non-technical related issues. The first has to do with acceptance: how will we convince people to adopt the ENS? Especially, we need to make sure that the benefits outnumber the concerns by proving the advantages of the service in a few very visible and popular domains. The second 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 entities. The profiles which we will store will be minimal, and will serve only to support 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 may become the enabling factor which will make possible for *knowledge-representation-on-the-web* to "shake the world" as it has never done before.

## References

1. Berners-Lee, T., Fielding, R., Masinter, L.: RFC 3986: Uniform Resource Identifier (URI): Generic Syntax. IETF (Internet Engineering Task Force) (2005), `http://www.gbiv.com/protocols/uri/rfc/rfc3986.html`
2. Pantel, P., Philpot, A., Hovy, E.H.: 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, pp. 438–439 (2006)
3. Monge, A.E., Elkan, C.: An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In: DMKD (1997)

4. Stoermer, H.: OKKAM: Enabling Entity-centric Information Integration in the Semantic Web. PhD thesis, University of Trento (2008), http://eprints.biblio.unitn.it/archive/00001389/
5. Hogan, A., Harth, A., Decker, S.: Performing object consolidation on the semantic web data graph. In: i3: Identity, Identifiers, Identification. Proceedings of the WWW 2007 Workshop on Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8 (2007)
6. Bouquet, P., Stoermer, H., Xin, L.: 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 (SWAP 2007), Bari, Italy, December 18-20 (2007), http://CEUR-WS.org/Vol-314/41.pdf
7. Bortoli, S., Stoermer, H., Bouquet, P.: Foaf-O-Matic - Solving the Identity Problem in the FOAF Network. In: Proceedings of the Fourth Italian Semantic Web Workshop (SWAP 2007), Bari, Italy, December 18-20 (2007), http://CEUR-WS.org/Vol-314/43.pdf
8. Jaffri, A., Glaser, H., Millard, I.: Uri identity management for semantic web data integration and linkage. In: 3rd International Workshop On Scalable Semantic Web Knowledge Base Systems, Springer, Heidelberg (2007)
9. Bouquet, P., Stoermer, H., Giacomuzzi, D.: OKKAM: Enabling a Web of Entities. In: i3: Identity, Identifiers, Identification. Proceedings of the WWW 2007 Workshop on Entity-Centric Approaches to Information and Knowledge Management on the Web. CEUR Workshop Proceedings, Banff, Canada, May 8, 2007 (2007), online http://CEUR-WS.org/Vol-249/submission_150.pdf ISSN 1613-0073
10. Bouquet, P., Stoermer, H., Mancioppi, M., Giacomuzzi, D.: OkkaM: Towards a Solution to the "Identity Crisis" on the Semantic Web. In: Proceedings of SWAP 2006, the 3rd Italian Semantic Web Workshop. CEUR Workshop Proceedings, Pisa, Italy, December 18-20, 2006 (2006), online http://ceur-ws.org/Vol-201/33.pdf ISSN 1613-0073
11. Kripke, S.: Naming and Necessity. Basil Blackwell, Boston (1980)
12. Bouquet, P., Stoermer, H., Cordioli, D., Tummarello, G.: An Entity Name System for Linking Semantic Web Data. In: Proceedings of LDOW 2008 (2008), http://events.linkeddata.org/ldow2008/papers/23-bouquet-stoermer-entity-name-system.pdf