Entity Name System: The Back-bone of an Open and Scalable Web of Data

Paolo Bouquet
DISI
University of Trento, Italy
bouquet@disi.unitn.it

Heiko Stoermer
DISI
University of Trento, Italy
stoermer@disi.unitn.it

Claudia Niederee
Forschungszentrum L3S
Hannover, Germany
niederee@l3s.de

Antonio Maña
Computer Science Department
University of Malaga, Spain
amg@lcc.uma.es

Abstract

Recognizing that information from different sources refers to the same (real world) entity is a crucial challenge in instance level information integration, as it is a pre-requisite for combining the information about one entity from different sources. The required entity matching is time consuming and thus imposes a crucial limit for large-scale, dynamic information integration.

An increased re-use of entity identifiers (or names) across different information collections such as RDF repositories, databases and document collections, eases this situation. In the ideal case, entity matching can be reduced to the trivial problem of spotting the same entity identifier in different information collections.

In this paper we propose the use of a Entity Name System (ENS) – as it is currently under development in the EU-funded project OKKAM – for systematically supporting the re-use of entity identifiers. The main purpose of the ENS is to provide unique and uniform names for entities for the use in information collections, so that the same name is used for an entity, even when it is referenced in different contexts.

Of course the creation of an ENS that can efficiently deal with entities on the Web scale raises scalability issues of its own. However, this paper focuses on the role of an ENS in contributing to the scalability of ad-hoc and on demand information integration tasks.

1 Introduction

In a very early note published in 1998[1], Tim Berners-Lee describes his vision of the Semantic Web as a global space for the seamless integration of countless semantic knowledge bases into an open, decentralized and scalable knowledge space. Much progress has been made since then to make this vision happen, but we must frankly admit that we are still far from such a reality. One of the main reasons seems to be that today the Semantic Web looks very much like a collection of "information islands" that are very poorly integrated with each other; and when some of these islands are linked, this is often the result of a lot of hard and time-consuming manual work.

Ideally, the integration of information islands into a global Semantic Web should be based on the practice of using Unique Resource Identifier[2] (or URI) for referring to any type of resource in RDF/OWL content. The key concept is that "[t]he global scope of URIs promotes large-scale *network effects*: the value of an identifier increases the more it is used consistently"[3]. Indeed, if two RDF graphs have two nodes labelled with the same URI, these two nodes can be collapsed, and the result is a bigger virtual graph where knowledge about that resource is automatically integrated. This key idea (which is perhaps the single most important difference between the Semantic Web and traditional knowledge representation in AI) has two important consequences: (i) on the one hand, we should avoid assigning the same URI to two or more different resources (URI collision), as they may introduce ambiguity and therefore "false positives" in information integration; (ii) on the other hand, we should try to avoid unnecessary URI aliases (i.e. associate arbitrarily different URIs with the same resource), as this would divide the Web of related resources by causing "false negatives" in information integration. Since we be-

[1] See `http://www.w3.org/DesignIssues/RDFnot.html`.

[2] See `http://www.w3.org/Addressing/`.

[3] See *Architecture of the World Wide Web, Volume One* (W3C Recommendation 15 December 2004) at `http://www.w3.org/TR/2004/REC-webarch-20041215/`.

lieve that the latter result is by foar more difficult to achieve, in the rest of the paper we will focus mainly on it.

To date, the good practice of associating the same URI with the same resource on the Web is not supported by any large-scale web infrastructure. This means that there is no easy and "standard" way for preventing the creation of URI aliases; as a consequence, a new URI is minted for the same resource any time a statement is made about it in different locations of the Web. This is particularly true for non-information resources (i.e. resources whose essence is not information[4]), including both universals (like classes and properties in an OWL ontology) and particulars (basically instances – of persons, locations, organizations, etc. – in an OWL knowledge base). This leads to the Semantic Web version of two well known problems in information integration:

- heterogeneity of vocabulary: the same concept (e.g. "person") or property (e.g. "first name") may be referred to through different URIs, and therefore may not be recognized as the same concept or property in two different vocabularies;
- entity recognition: the same real world object (e.g. "Florence") may be assigned different URIs in different RDF repositories, and therefore may not be recognized as the same entity.

While the first issue is widely recognized and investigated[5], for a long time the second was largely neglected in the Semantic Web community, though it received – and is receving again – a lot of attention in the database community (under the headings of record linkage, data deduplication, entity resolution, etc. [8, 5]). However, this concentration on schema issues looks like a serious strategic mistake. Indeed, without underestimating the role of vocabularies, we think that the Semantic Web will not happen when we will have a large number of interlinked vocabularies, but when we will have a large number of data collections (mainly in the form of RDF repositories) which provide decentralized and independent information about *the same entity*. And the most powerful way for knowing that information is about the same entity is that such an entity is associated with the same URI across these different repositories.

As a possible solution to this situation, in this paper we present the concept of the **Entity Name System** (ENS), a web-scale infrastructure for supporting the reuse of pre-existing URIs for any type of entity across decentralized and independent RDF repositories. We will argue that the ENS can become the backbone of the Semantic Web, as it provides an infrastructure for the creation of RDF content which can be easily integrated via simple graph merging (i.e. without any preliminary step of "entity matching" to determine whether two URIs refer to the same world object).

The rest of the paper is structured as follows. Section 3 discusses the role of entity identifier re-use for instance level information integration and its impact on scalability in more detail and presents exemplary applications of our approach. In section 2 the OKKAM approach to entity identifier management and the architecture of the planned ENS is discussed. Issues to be considered in realizing an ENS are discussed in section 3. Our approach is related to other work in the area entity identifier management in section 4. Finally, the paper concludes with a summary and ideas for future work in section 5.

2 OKKAM Approach and Architecture

The key idea behind the proposal of an ENS is that the Semantic Web can become an open and scalable space for publishing knowledge (in the form of RDF data) only if there will be a reliable (and trustworthy) support for the reuse of URIs. Therefore, at a very general level, the core functionality of the ENS can be characterized as follows: given any representation of an entity (e.g. a bag of keywords, a paragraph of text, a collection of key-value pairs, a graphical depiction, and so on), decide if a URI for this entity is already available in an entity repository (using some method(s) for *entity matching*); if it is, then the ENS will return its URI (or at least a ranked list of candidates), otherwise it will issue a new URI which will be stored in the ENS repository.

To achieve this result, a lot of technical and non-technical issues must be addressed. In this section propose a concrete architecture for implementing the ENS on the Web; in the next section, we will discuss some of the corresponding issues.

2.1 ENS: the OKKAM approach

As we have argued in [2], issues of entity identification are optimally solved a-priori, across data sources and formats. Instead of creating RDF repositories in which the same real-world entity is denoted by two or more different URIs, and then trying to reconcile these URIs, we should aim at enabling any application which produces RDF content to reuse a globally unique URI for that resource from the outset.

The positive effects are evident. Instead of using one of the many possible names for an entity[6], a uniform elec-

[4]See `http://www.w3.org/TR/2004/REC-webarch-20041215/#identification`.

[5]See e.g. [6] for a recent survey of approaches and tools for schema-level alignment of ontologies.

[6]The interested reader is referred to the seminal philosophical discourse

tronic surrogate is used. The local effect within a single system is that ambiguities of references to entities in metadata can be eliminated to the largest part already at creation time. The global effect is that: (i) information integration is largely reduced to schema level integration, as entity identifiers provide large parts of data-level integration for free (besides dealing with conflicting and redundant data in different collections), and (ii) completely new hyper-structures are possible that link between different entities and between artifacts and entities via the shared entity identifiers.

Optimally, such a global identifier for every entity referenced in a data source is used throughout *all* records/terms/statements that refer to this entity, in every data source referring to this entity, and in (external) content such as websites or other documents. This leads to the possibility to relate and integrate – without additional efforts – textual and multimedial content referring to a specific entity. This becomes more and more relevant taking into consideration the fast pace of development in multimedia libraries, as can be seen in current services such as YouTube[7] or Flickr[8].

As a consequence, the adoption of such an approach can facilitate the realization of many more advanced services and enable various innovative "entity-centric" applications, i.e. applications that are centered around the notion of an entity.

2.2 The OKKAM Entity Name System Architecture

In OKKAM, the ENS is implemented as a federated architecture, which is illustrated in Fig. 1. Each node in the cloud labelled OKKAM*PUBLIC* is internally organized according the architecture depicted in Fig. 2 and described in full detail in [13]. Additionally, the architecture foresees what we call a *local node*, which is a smaller-scale OKKAM node with the additional functionality that makes it operate like a caching proxy. This private node is meant for use cases where institutions have to manage entities (or information that characterizes entities) which need to be physically stored on the institution's infrastructure. Such a local node can accomodate highly sensitive information, or simply allow for cases where vast amounts of entities have to be managed which are only of interest inside the institution and would thus represent "pollution" of the public global service.

The prototype which is currently under developmend is the third evolution of an ENS since it was first presented [3] and subsequently implemented as a single node, analyzed and used in experiments [13]. The aim of this new prototype to be finished in 2008 is to provide a more complete set of distributed ENS functionality, an adaptive matching layer, and vastly improved storage.

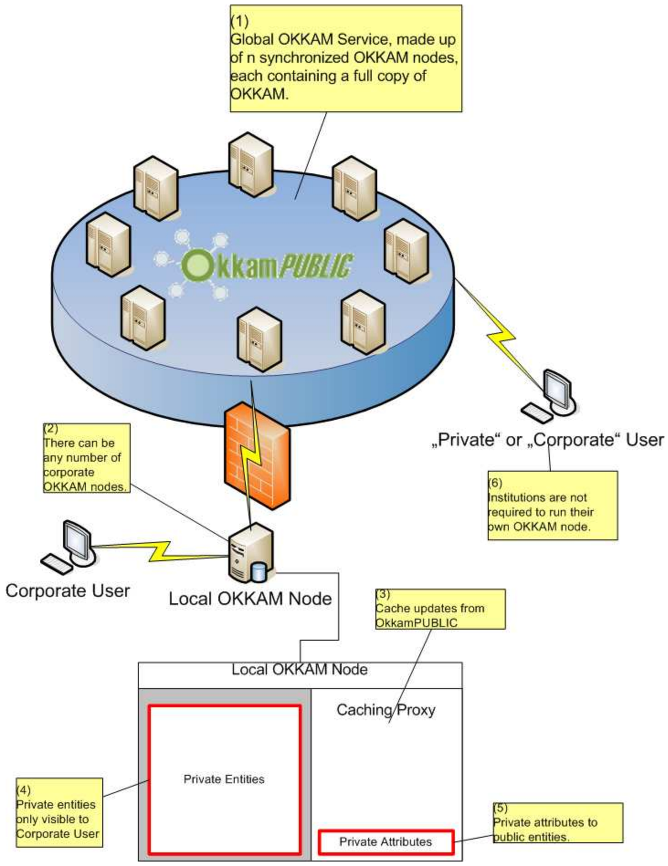

Figure 1. Global, distributed architecture of the OKKAM ENS

What is illustrated in Fig. 1 is a distributed system that is fully in line with the distributed nature of the (Semantic) Web. 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[9], which prevents accidental generation of duplicates and thus also enables their use as primary keys e.g. in relational data sources (i.e. avoiding URI collision).

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.

about naming by Saul Kripke [10].

[7] `http://www.youtube.com`

[8] `http://www.flickr.com`

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

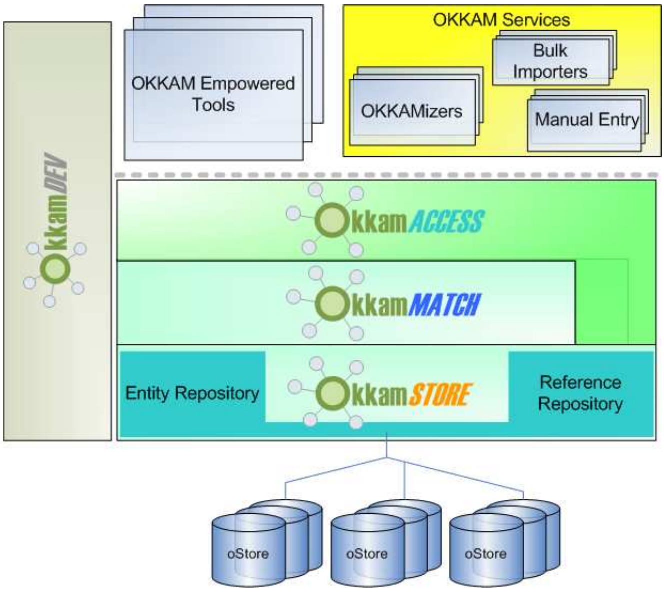

Figure 2. One node of the public OKKAM infrastructure

3 Issues in Scalable, Entity-centric Information Integration

3.1 Dealing with Entity Types

The type of an entity and a possibly associated schema for its representation (description) raise further issues to be considered in the creation of an EI Management solution. This becomes relevant if EI Management is to be implemented for more than one type of entity.

An entity repository with a strong notion of typing is expected to increase efficiency and effectiveness of entity identifier retrieval, because entities can be managed in virtually or physically separate repositories according to their types, and type-specific matching approaches can be implemented. This raises the challenge of finding the right granularity and the right set of types for organizing the repositories. Furthermore, the selected grouping of entities into types has to be agreed within the community (e.g., bridging differences in conceptualization, naming). A promising starting point for this is a set of top level types such as "person", "organization", and "event", as they can be found as top level classes in some Upper Level Ontologies (e.g., SUMO [12], DOLCE [7]). A second obstacle in entity typing in EI Management is that the information provided by the user of the EI Management may –in some cases– not be sufficient to decide about the type of the searched entity. Here, it has to be considered that the "user" might as well be another application, which makes it far more complex to request further clarifying information from this "user", when required.

An alternative, more flexible solution with respect to typing, is to manage all entity representations together and keep available type information as links, e.g., to ontologies, or other schemata. This avoids the need for an a-priori agreement on a set of entity types and copes with the cases where there is not enough information to determine it. However, it is foreseen that entity identity decisions become more complex and less efficient in this case.

In our solution we will start with the "top level types" solution sketched above.

A second, although related topic, is the use of schemata for the representation of the entities. For the EI Management solution it would be easiest to have a fixed schema for the description of every type of entity and to also use this schema when issuing identifier requests. Since it is not the idea of the EI Management to collect as much information as possible about each entity, this schema should include the attributes that are most adequate for the identification of the respective entity. However, we expect the EI Management to operate in a highly heterogeneous setting, which requires more flexibility with respect to the representation schema. For the usability in different situations it would be best to enable the user to use the attributes he has at hand for querying the entity repository (following the respective local schema, e.g., the employed metadata schema).

We will experiment in our solution with a two level approach. On the lower level (storage and matching candidate retrieval) we will make use of a core schema for every entity type to enable very efficient and scalable solutions. This schema will be dynamically adapted, based on data learned from the usage of the entity repository[10]. This will enable efficient retrieval of matching candidates. On the higher level, we will cope with the translation of incoming queries into core schema requests (schema mapping) as well as with analyzing the matching candidates returned by the lower level. The higher level will have much more flexible notion of schema and will also support the more user-friendly solution of using local schema information.

3.2 Repository Maintenance

The core EI management functionality has been sketched above (entity identity decisions, issuing entity identifiers, managing entity identifiers and entity representations). The purpose of entity representation is to enable entity identity decisions. An initial representation is created when the entity identifier is issued from the information that was provided as part of the respective request. For the update and extension of the information managed for the individual

[10] Such a solution also has implications for the entity repository maintenance processes.

entities, we foresee the following processes: a) collection of information that is provided when further requests for the same entity are encountered (using different information about the entity, b) analysis of the usefulness of the stored entity information, c) consideration of the age of the considered information, and d) (only in rare cases) manual change of entity information via adequate user interfaces. The methods that will be developed for entity representation update and extension based on a), b) and c) are part of the repository maintenance processes, which have the goal to maintain the quality of the repository on the long run.

Furthermore, repository maintenance processes include repository purging. Incrementally, entity representations will change, which means that the setting in which an entity identity decision has been originally taken changes as well. This suggests that it makes sense to revisit such decisions as part of what we call a repository purging process. Such a process revisits identity decisions, i.e. it checks if given the current status of information in the repository entity matching would still support the same entity identity decisions. As a result of such a process it might be detected that two entity representations (with separate identifiers) actually refer to the same entity requiring corrective actions (a complete revision of the original decision is not possible since both identifiers might already be in use outside the EIM system). The process might also detect evidences for the fact that two real world entities have been by mistake or lack of sufficient information been marked as identical. The revision of earlier entity identity decisions of course has an impact on the identifiers already issued by the repository. As a basic principle we foresee that identifiers are never deleted (besides in exceptional cases). Thus an equivalence statement will be added which couples the two entity representations.

The revision of identity decisions does not only have impact on the repository, but also on the users of the entity identifiers. Thus, it has to be considered whether it makes sense to provide an option to inform the users of the respective entity identifiers about the change. This will depend on the required effort for enabling this (e.g., the number of users) and the desired degree of autonomy and independence between the repository and the identifier users.

3.3 Trust, Privacy and Ownership

Systems as the one proposed in this paper frequently contain objects with heterogeneous security and privacy requirements, depending on the application scenario, that pose important challenges on the underlying access control and security mechanisms.

In fact, it follows the paradigm of a distributed, heterogeneous and large-scale system with highly dynamic security requirements, a large number of users and very crucial security and privacy requirements. In particular, the need for the EI Management to be very open, the vast variety of applications that can be supported by the presented services, and the nature of the information stored in the Entity Repository, together with the possibilities for misuse that the existence of such a repository creates, require the design and development of very flexible security mechanisms [15].

One of the main pillars of these security mechanisms is access control (supported by identification and authorization). Paradoxically, access control in distributed systems often relies on centralized security administration. Centralized control has obvious but important disadvantages: (i) the control point represents a weak spot for security attacks and fault tolerance, (ii) it reduces system performance because it introduces a bottleneck for request handling, and (iii) it usually enforces homogeneous access control schemes that do not fit naturally in heterogeneous user groups and organizations.

On the other hand, systems for distributed security administration still have open problems. Solutions proposed so far do not provide the flexibility and manageability required. Several access control models have been introduced in the literature to fit different access control scenarios and requirements. Some schemes have also tried to integrate different models in a unified framework. These approaches represent significant advances over traditional single-policy systems but, unfortunately, are still constrained by the underlying models and do not provide the necessary flexibility.

We believe that a more general approach is needed in order to be used in these new environments and in particular in the one presented in this paper. For example, in the referred situations, groups are an artificial substitute for a more general tool: the attribute. In fact, groups are usually defined based on the values of some specific attributes (employer, position, access level, etc). Some attributes are even built into most of the access control models [14]. Similarly is the case of the user element; the identity is just one of the most useful attributes, but it is not necessary in all scenarios and, therefore, it should not be a built-in component of a general model.

Finally, in distributed computing environments, there are many different situations where it is desirable that the owner of each resource is able to retain the control over it and to change the access policy dynamically and transparently regardless of the location where the resource is stored. This property is called originator-retained-control [11]. Ownership in an EIM system is however not precisely related to an entity as a whole: in such a collaborative environment also individual pieces of information about an entity (such as an individual descriptive label) could be considered objects that underlie ownership. Thus, it is necessary to establish models that operate on a very fine-grained level of detail.

3.4 Scalability

Scalability in the context of this paper can be viewed from at least two perspectives:

1. Bringing the Semantic Web to a scale comparable to that of the WWW today, and
2. evaluating the scalability issues of an ENS itself.

While the first has been illustrated in the introduction, with the claim that one crucial factor of scaling the Semantic Web is to minimize proliferation of entity identifiers and the resulting matching effort, in this section we want to look at the second issue.

It is clear that a global system for entity identifier management faces considerable challenges with respect to scalability. At first, it might appear that the performance and scalability required to provide such a service is immense, and indeed, they can not be ignored. However, three main use-cases can be identified which affect the system in different ways, and are expected to keep scalability demands manageable. We will try to relate to statistics known about the popular Wikipedia system[11] for comparison, which serves a rough average of $30,000$ document requests per second[12].

Creation of new entity identifiers. This use-case is obviously a demanding one, as it causes write access to the underlying data storage which is inherently costly. The question is how many new entries are going to be published per second, and the answer is not easily quantifiable. As a starting point, let us consider creating an identifier for every person on the planet[13]. If we assume an unrealistically great success of the approach, within one week we would create 6.6 billion entries, resulting in a load of about $11,000$ requests per second which should be in a manageable range for modern cluster architectures. We believe this to be an exaggerated example, because many of the "typical" types of entities (people, locations, etc.) have a rather high number to begin with, but do not show explosive and unmanageable growth rates.

Entity search. Search on a massive amount of records or documents is not a new problem, and solutions have been evolving from research projects to commercial products. Large players in search technology are impressively demonstrating what already today is feasible. Wikipedia – even being a non-profit organization with somewhat limited resources – manage to serve $30,000$ document requests *per second*. We believe that by deploying a highly optimized storage and query infrastructure which advances the state of the art in information retrieval and high-performance matching it is possible to deal with the amount of queries that are to be expected.

Usage of entity identifiers for information integration. It is important to point out that the use of a centrally issued entity identifiers in metadata records, but also during information integration processes, does not cause any processing load on the EI Management system. Once such an identifier is stored in a metadata record, it "leaves" the context of the management system and becomes an independent datum that can serve by itself, and without any backing of the management system, as a pivot or anchor for information integration. An integration process that relies on entity identifiers as pivots is thus outside the scope of this scalability discussion, as it does not lead to read or write request in the system.

We are aware of the great importance of the scalability aspect, but also believe that the points above help clarify where the real challenges lie. Additionally, we think that the efforts described in Section 5 about future work will credibly illustrate that we are not basing our work on unrealistic assumptions.

4 Related Work and Projects

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

The first is the *Linking Open Data Initiative*[14], 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 community has made a huge effort to link a significant amount of data, their approach depends on specialized, data source-dependent heuristics[15] to establish the `owl:sameAs` statements between resources, and it requires the statements to be *stored* somewhere, along with the data. As we said in the introduction our main concerns with this approach (without the ENS) are the following: first, in most Web scenarios, we don't see standard web users making an effort to create `owl:SameAs` statements for their data; second, an error in an identity statement might have long ramifications on the entire Web of Data;

[11] `http://en.wikipedia.org/wiki/Wikipedia:Statistics`

[12] `http://hemlock.knams.wikimedia.org/~leon/stats/reqstats/reqstats-yearly.png`

[13] The world population is at an estimated 6.6 billion in February 2008, according to the U.S. Census Bureau(`http://www.census.gov/ipc/www/idb/worldpopinfo.html`)

[14] `http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

[15] `http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/EquivalenceMining`

finally, reasoning over massive numbers of `owl:sameAs` statements in distributed ontologies is computationally a complex and highly expensive task, which may lead to the conclusion that these linked data are more suitable for *browsing* than for reasoning or querying, and thus do not fully attempt to realize the vision of the Semantic Web as a large, distributed knowledge base.

The second is presented in Jaffri et al. [9]. In their work resulting from the ReSIST project, these authors recently came to a conclusion similar to the the one we had already expressed in [3, 4], 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[16]. 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 above.

5 Conclusions and Future Work

Of course there are various challenges to be solved and methods to be developed for making scalable, entity-aware information integration a reality. This especially refers to the creation of a flexible, efficient and scalable ENS. We will address these challenges in the next versions of the ENS, where the special focus of the next version will be on the following aspects:

- Advanced entity matching methods that can deal with highly heterogeneous entity description schemata: the entity identifier management system has to be able to decide whether an ID already exists, given the information a user or application presents to the system. The type of presented information as well as the type of considered entity may vary widely.

- Development of a well-defined model of "entity" and "entity identity": sound foundations are required as a basis for decision making on identity within the system, e.g. for adequately resolving situations where entities are variants, parts, versions of each other.

- An approach and methods for the management of the entity lifecycle: since it is a requirement that entity IDs persist for a very long time in the system, methods are required for dealing with evolving the contents and relevance of entity information, rules for deletion and update, methods for merging entity IDs, etc., which we summarize under the term *entity lifecycle management*.

[16] see e.g. Kripke [10]

In addition to these research challenges there are also further technical and organizational challenges to be resolved. This includes creating an sufficiently large initial population of the entity repository to make it attractive for adopters (and fostering global identifier re-use). For the same purpose, it is also important to create some attractive entity-centric applications and to establish flexible relationships with other identifier management solutions.

6 Acknowledgements

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`.

References

[1] 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`.

[2] P. Bouquet, H. Stoermer, and B. Bazzanella. An Entity Naming System for the Semantic Web. In *Proceedings of the 5th European Semantic Web Conference (ESWC2008)*, LNCS, 2008. to appear.

[3] 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.

[4] 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.

[5] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[6] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[7] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with dolce. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 166–181, London, UK, 2002. Springer-Verlag.

[8] H. Garcia-Molina. Pair-wise entity resolution: overview and challenges. In P. S. Yu, V. J. Tsotras, E. A. Fox, and B. Liu, editors, *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006*, page 1. ACM, 2006.

[9] 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.

[10] S. Kripke. *Naming and Necessity*. Basil Blackwell, Boston, 1980.

[11] J. Lopez, A. Mana, E. Pimentel, J. M. Troya, and M. I. Y. del Valle. Access control infrastructure for digital objects. In *Information and Communications Security, 4th International Conference, ICICS 2002, Singapore, December 9-12, 2002, Proceedings*, pages 399–410, 2002.

[12] I. Niles and A. Pease. Towards a standard upper ontology. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9, New York, NY, USA, 2001. ACM.

[13] H. Stoermer. *OKKAM: Enabling Entity-centric Information Integration in the Semantic Web*. PhD thesis, University of Trento, January 2008. `http://eprints.biblio.unitn.it/archive/00001389/`.

[14] M. I. Yagüe, M. del Mar Gallardo, and A. Mana. Semantic access control model: A formal specification. In *Computer Security - ESORICS 2005, 10th European Symposium on Research in Computer Security, Milan, Italy, September 12-14, 2005, Proceedings*, pages 24–43, 2005.

[15] M. I. Yagüe, A. Mana, and F. Sanchez. Semantic interoperability of authorizations. In *Proceedings of the 2nd International Workshop on Security In Information Systems, WOSIS 2004*, pages 269–278, 2004.