Foaf-O-Matic - Solving the Identity Problem in the FOAF Network

Stefano Bortoli[1], Heiko Stoermer[1], Paolo Bouquet[1], and Holger Wache[2]

[1] University of Trento,
Dept. of Information and Communication Tech.,
Trento, Italy
{bortoli, bouquet, stoermer}@dit.unitn.it
[2] University of Applied Sciences Northwestern Switzerland
School of Business
Brugg, CH
holger.wache@fhnw.ch

Abstract. An issue that is equally arising both from social networks and the Semantic Web is the fact that, without the consistent use of the same identifier for an object across systems, it is unnecessarily hard to perform information integration. We are addressing this issue where both fields intersect, namely in the context of FOAF profiles, which describe the social network of a person with Semantic Web technology. We have developed a new, state-of-the-art web application for the generation of FOAF profiles, which has the integral characteristic of making use of the Okkam Infrastructure. This infrastructure enables the systematic, global use of unique identifiers for entities, and thus aims to solve the identity problem that exists in FOAF today.

1 Introduction

In the past few years, big efforts have been performed in the context of Semantic Web aiming at developing tools and solutions focused on mapping among ontologies and taxonomies, distributed reasoning, automatic knowledge extraction, building ontologies starting from databases and so on. In some sense, the Semantic Web relies on a silent underlying assumption: users throughout the Web will realize their sets of RDF assertions and then share them with others. Each set of RDF assertions represent a graph, where the nodes represent the resources and the edges correspond to properties. The integration between different graphs is obtained by merging the graphs by the means of collapsing the nodes describing the same resource. This process results in a graph in which the information regarding a resource is in some sense the sum of the information contained within the initial graphs. The integration among different graphs relies on the possibility to identify nodes describing the same resource.

When defining RDF assertions, the described resources are identified by Unique Resource Identifiers (URI). Nodes with the same URI in different graphs refers to the same resource, and so they may be collapsed when merging the

graphs they belong to. However, it is unclear how different developers working on their own ontologies and describing the same resource may be able to consistently use the same URIs for them. This is, in our opinion one of the biggest issues affecting the Semantic Web: the lack of "reusing and sharing" already defined identifiers for entities. Indeed, the merging of graphs relies on collapsing identifiers referring to identical resources, or, equivalently, assertions of identity binding different identifiers describing the same resource[3]. In this way, we assist on a proliferation in the creation of identifiers describing the same resource. We call this problem "*Pirandello's identity problem*", a more precise description of which is presented in section 2, and whose actual relevance in the area of social networks can be inferred from [5].

Under these premises, in this paper we try to prove the benefit arising from the use of Okkam [2] as means for sharing and reusing identifiers. In order to perform this experiment we chose to deal with one of the most successful applications of the Semantic Web, namely Friend-Of-a-Friend (FOAF). The experiment lead to the development of a new application integrating Okkam and providing functionalities for the creation of FOAF RDF profiles.

2 FOAF and Pirandello's Problem of Identity

Luigi Pirandello (1867-1936) was an Italian dramatist, novelist, and short story writer awarded the Nobel Prize in Literature in 1934. He is author of "One, No one and One Hundred Thousand" [7], a novel in which the protagonist discovers how all the persons around him have constructed in their mind a specific view of him, and how none of these views correspond to the image he has of himself. The problem experienced by the protagonist of the book of Pirandello can be used as a metaphor to explain the nature of the problem we are aiming to solve.

In a sense, every time that a new identifier is created, a new view about a resource (entity) is created. This multiplication of identifiers makes the concept of identifier itself weaker. Indeed, if an identifier for a resource is recognized by a single agent (the creator), or simply within a limited context, then we can state that this resource does not exist, or cannot be recognized for what it is, in an external context. The URI identifying a resource has, or should have, the good property of being unique, but, as long as it can be created arbitrarily, it cannot fulfil this property. Thus, creating different instances characterized by different URIs identifying the same resource leads to a parting of the associated information, and therefore a consistent decrease in the capability of making inferences and extracting knowledge. It is worthy to underline that this problem has a subtle difference from the "identity crisis" problem analyzed by Pepper [6]. Indeed, Pirandello's *identity problem*, as we mean it, describes the lack of reuse of identifiers referring to the same resource, and the weak identifying property of an arbitrarily created URI.

In order to show the concreteness of the described problem, we chose to analyze a limited context referring to a real world application: namely the Friend-of-

[3] This is, for example, the approach pursued by the Linked Data Initiative [1]

a-Friend (FOAF) project[4]. The FOAF initiative regards the definition of a set of specifications and tools based on the W3C's RDF language [4] that allow agents (people, organization, groups etc.) to describe themselves, their place of work, their main interests, education institutes etc. Furthermore, the set of properties associated to a FOAF agent are conceived to state some relationship involving other agents. The most important and used is the "*foaf:knows*" object property relating FOAF Person resources. In simple words, by means of this object property it is possible to state who is friend of whom and share this knowledge on the web in a machine readable way.

The vocabulary of FOAF is reasonably expressive, although still in evolution, and allows to express different types of information describing a person [5].

Analyzing the set of properties describing a FOAF *person* entity, it becomes clear that the best identifier *currently* available is the unique code obtained by encoding a person's email address in the field (*/foaf:mbox_sha1sum*). Indeed, an email address is uniquely identifying a mailbox of a person. Furthermore, often people use the same email address for long periods of time and this fact make the email address useful to identify persons along this period.

Any FOAF file describing a person represents an RDF graph. Every single graph is supposed to be merged with other graphs collapsing the nodes identifying the same person. Namely, if in two graphs somewhere the same unique code derived by the mail address is used, then both graphs contain some kind of information about the same person, therefore the graphs can be merged enlarging the network of "friendship". By means of this procedure it is possible to build a bigger graph containing all the information stated by the respective social network.

Analyzing superficially this process, everything seems to be at the right place, but going a little bit deeper some problems arise. The problems are related mainly to the weakness of the use of the email address based code as identifier. Indeed, an email address is not a good identifier for the following reasons:

- people change email address (change work/study institution, choose better provider, drop over-spammed[6] email address, etc...)
- people use more than one email address depending on the context of use (work, on-line gaming and shopping, 'night activities', family and friends relationship, etc...);
- email addresses can act as proxies for more than one person.

The facts listed above raise the following problem: different actors could use different email address to identify the same person (agents). Thus, a complete merging of all the information regarding a person is no more even possible. Despite the analyzed context is pretty simple and circumscribed, it reflects the more general Pirandello's identity problem affecting the semantic web evolution.

[4] The web page of the project is: `http://www.foaf-project.org`

[5] For more detail about the FOAF vocabulary see *http://xmlns.com/foaf/0.1/*

[6] over-spamed means that this address is a constant target of spam email

The weakness of the identifier generated on the base of the email address can be tackled and resolved by means of adding globally unique identifiers that are not dependent from the context of use. Thereby, there is the need of a tool that supports the creation of this kind of identifiers, namely OKKAM.

3 The Okkam*PUBLIC* Infrastructure

The work described in this paper relies on the existance of the OKKAM infrastructure, the initial idea of which was described in more detail in [3, 2]. As illustrated in Figure 1, at the heart of this infrastructure there is the central repository for entity identifiers, called OKKAM*PUBLIC*[7]. This repository can be imagined like a very large catalog, where semi-structured descriptions of entities are stored and associated to globally unique identifiers for these entities. It furthermore provides the functionality to add entities and their descriptions to the repository that have not existed there so far, and to retrieve their OKKAM identifiers for use in information systems.

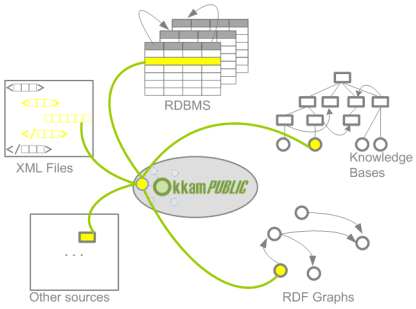

Fig. 1. Overview of the global OKKAM vision.

Figure 2 illustrates the standard use-case for the *okkamization*[8] of content, namely to query OKKAM*PUBLIC* for the existance of the entity at hand. This would usually be achieved through functionality provided by a client application – in this case FOAF-O-MATIC – which accesses the OKKAM*PUBLIC* 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

[7] This service is currently under development at the University of Trento, and will be opened for public access in the near future.

[8] 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.

entity in OKKAM and thus cause an identifier for the entity to be issued and used as described before.

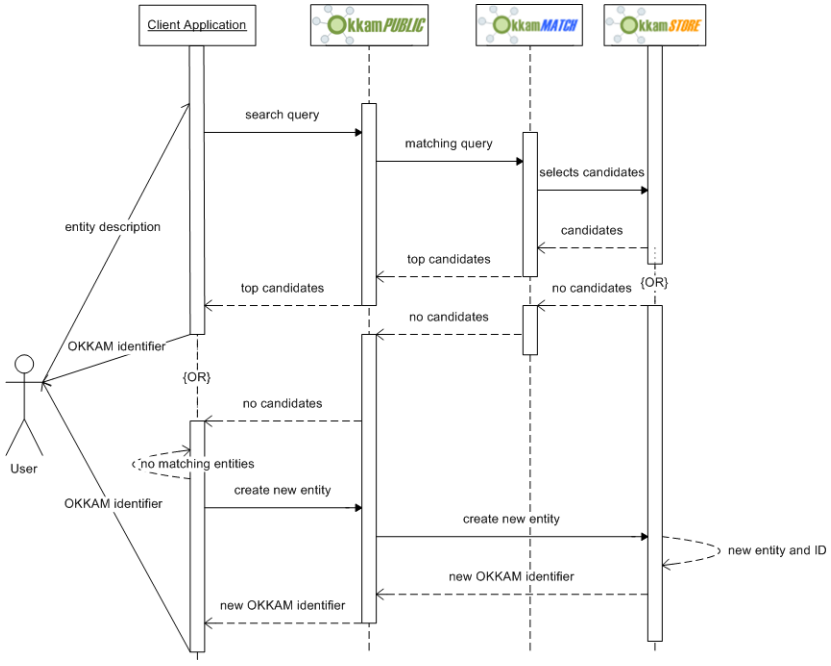

Fig. 2. Sequence diagram of the OKKAM standard use case.

The large-scale, global service OKKAM*PUBLIC* provides for the entity repository and a service infrastructure so that tools and applications can make use of this new technology. The current version of OKKAM*PUBLIC* is a prototypical implementation of parts of a larger multi-tier architecture, namely a non-distributed version of the storage component OKKAM*STORE* which in a later phase will move to a distributed layout, a preliminary version of the matching component OKKAM*MATCH* which performs the search for entities, and a subset set of the developer API and toolkit OKKAM*DEV* which is available[9].

The mechanisms inside OKKAM which perform the matching between entity descriptions provided by the user or agent and the existing descriptions stored in the repository, display some specifics which should be mentioned at this point. One of the main characteristics of OKKAM is that the description of an entity, which is necessarily used to distuingish this entity from all others in the repository, does *not* follow a fixed schema, i.e. OKKAM is specifically not something like a knowledge base of entities; consequently, OKKAM is not providing an ontological formalization of which attributes an entity has. The way to describe

[9] `http://www.okkam.org`

entities is extremely flexible and semi-structured, realised by way of key/value pairs which can contain arbitrary strings. The reasons for this decisions have been laid out in [3, 2], and basically go back to the point that there is an infinite variety of ways of how to model domains, for which reasons we decided to stay completely domain independent. As a consequence, the matching algorithms in OKKAM*MATCH* can take as input *any* kind of description of an entity, e.g. the set of properties and values inferred from an ontology or RDF graph, and match it against existing data. This is how we achieve OKKAM support without any dependence on, or knowledge of, an underlying schema.

4 Foaf-O-Matic – A Solution Approach

In the previous sections we have illustrated that what is missing in a FOAF Person description is a unique and sole identifier.

Thus, the problem described in Sect. 2 looks to be a perfect example of a real-world case in which OKKAM could play an important role in terms of information integration, enhancing the merging of RDF graphs describing the same person or its social network.

The approach applied for tackling the analyzed problem is to provide a tool allowing users to create/integrate FOAF person descriptions with identifiers contained in, or generated by, OKKAM. Thus, what is needed is a new application extending the functionalities provided by the foaf-a-matic application[10]. In order to underline the historical relation with the former application, this new web-based tool has been named FOAF-O-MATIC[11] (with the 'O' underlining the integration with OKKAM.)

It is important to notice that the aim of creating the FOAF-O-MATIC application is not only to replace the slightly 'obsolete' foaf-a-matic application and providing a pretty layout and new description fields. The focal point of the new application is to allow users to integrate OKKAM identifier 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 new layout of the application is given in figure 3. As it is possible to see from the figure, the main layout is split in two columns: the left one for the *foaf:PrimaryPerson* description, and right one for the friend management. On the top of this two columns facilities to upload already defined FOAF files are presented. At the bottom, a "generate FOAF" button is present that trigger the generation and visualization of the FOAF file in a text area.

Without going too much into details, the FOAF-O-MATIC is meant provide the following set of functionalities:

[10] `http://www.ldodds.com/foaf/foaf-a-matic` — The foaf-a-matic is a tool that allow the definition of foaf rdf description by means of a simple form fulfillment.

[11] `http://www.okkam.org/projects/foaf-o-matic/`

Fig. 3. FOAF-O-MATIC screenshot.

- **Upload a FOAF file**. This functionality is meant to allow the upgrade of already defined FOAF descriptions and enhancing it with OKKAM identifiers. The file can be loaded providing either its Web URL, loading the file from the file system as is possible to see in the area marked with 1 in figure 3.
- **Describe the *foaf:PrimaryPerson* aka 'yourself'**. This functionality supersedes foaf-a-matic by providing of a wider choice of description fields some under testing FOAF properties. For a matter of dimension, the input for has been split in three collapsible panels presenting in the top part the standard description fields, in the middle part some extra information fields (i.e. *birthday*), and in the bottom part some *chat-id* related information fields (i.e. *yahooChatId*). A view of this part of the application is presented in figure 3 in the area marked with 2.
- **Add and describe friends**. This functionality is meant to allow users to provide a description of the friends they want to add to their social network. The information provided will be used to inquire OKKAM and retrieve a list of candidate entities corresponding to the described friends. If no entities will be found in OKKAM a newly created entity identifier will be provided. If none of the candidate entities match the user requirement in terms "recognition" a new identifier will be provided as well. "Okkamized" entities[12] will be marked in a special way. A view of this part of the application is presented in figure 3 in the area marked with 3 and 4. Notice that an OKKAM identifier is now part of the description of the described friend.
- **Select one Okkam entity for each described person**. This functionality is meant to allow the user to choose which is the entity representing the described person among the one matching such description within OKKAM, if any. The chosen entity identifier is used in the definition of the RDF FOAF file as value of *rdf:about* attribute of the described person. The list of candidate OKKAM entities is presented in a pop-up panel. The user can select the correct entity by pressing the "Select" button associated to the entity, or to state that none of the retrieved entities correspond to the describe person by pressing the button "None".
- **Retrieve the new FOAF description**. The FOAF RDF description containing the informations provided by the used is presented in a text area below the description areas. The FOAF RDF description containing the information provided and integrating an OKKAM identifier where chosen, is generated every time the "generate FOAF" button is pressed. The content of the file reflect the present state of the description provided by the user.

4.1 Foaf-O-Matic Development Framework

The framework used for the development of FOAF-O-MATIC is ICEFaces[13] open source project. ICEfaces is the most widely distributed enterprise Ajax[14] framework on the market today, providing a rich library of Ajax components. The

[12] entities which has been assigned an OKKAM identifier

[13] `http://www.icefaces.org/`

[14] Asynchronous JavaScript and XML - `http://www.ajaxprojects.com/`

main benefit of Ajax is that it gets rid of the usual submit/reload mechanism of Web forms and enables the creation of very user-friendly interfaces comparable to modern desktop windowing systems.

The primary goal behind the ICEfaces architecture is to provide a familiar Java Enterprise development model, and completely isolate them from the complexities of low-level Ajax development in JavaScript. The key to the ICEfaces architecture is a server-centric application model, where all application logic is developed in pure Java, and executes in a standard Java Application Server runtime environment.

The ICEfaces Framework is an extention to the standard JSF[15] framework, with the key difference in ICEfaces relating to the rendering phase. In standard JSF, the render phase produces new markup for the current application state, and delivers that to the browser, where a full page refresh occurs. With the ICEfaces framework, rendering occurs into a server-side DOM and only incremental changes to the DOM are delivered to the browser and reassembled with a lightweight Ajax Bridge.

5 Future Work and Conclusion

In this paper we have presented Foaf-O-Matic, an extended service for the creation of FOAF profiles, which relies on the Okkam infrastructure for issueing the "friends" with globally unique identifiers, and thus solving a-priory some of the issues of social network applications illustrated for example in [5].

The application we propose within the FOAF context is only one example of the potential application on top of an infrastructure – Okkam – which appeases the Pirandello's identity problem affecting the semantic web. But much more applications can also benefit from this infrastructure, in fact each application domain where some information about the same "thing" is distributed over different platforms is a candidate for Okkam improvements.

For the next steps we plan to extend Foaf-O-Matic in order to get some experience with Okkam and its matching algorithms. The benefit of the FOAF application is that there are many FOAF files distributed over the Internet which provide a good training base for the matching algorithm. With the FOAF application we want to tune Okkam's and Foaf-O-Matic's algorithms. With that experience we explore further application and improve Okkam over time and application domains. Also a scalable architecture with a fuzzy entity identification are subject of investigation.

The Okkam infrastructure itself will experience a great boost in the course of a European FP7 Integrated Project to start in early 2008 – consequently named Okkam – which has the aim and the means to implement the infrastructure briefly illustrated in Sect. 3 at a very large scale.

More information will be made available at `http://www.okkam.org`.

[15] JavaServer Faces - `http://java.sun.com/javaee/javaserverfaces/`

References

1. Tim Berners-Lee. Design Issues – Linked Data. Published online, May 2007. `http://www.w3.org/DesignIssues/LinkedData.html`.
2. Paolo Bouquet, Heiko Stoermer, and Daniel 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.
3. Paolo Bouquet, Heiko Stoermer, Michele Mancioppi, and Daniel 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.
4. Patrick Hayes. *RDF Semantics*, February 2004. http://www.w3.org/TR/rdf-mt/.
5. Tim O'Reilly. *The Social Network Operating System*, October 2007. online `http://radar.oreilly.com/archives/2007/10/social_network_operating_system.html`.
6. Steve Pepper and Sylvia Schwab. Curing the web's identity crisis: Subject indicators for rdf. Published online., December 2003. `http://www.idealliance.org/papers/dx_xml03/papers/05-01-05/05-01-05.pdf`.
7. Luigi Pirandello. *One, None and a Hundred Thousand*. E. P. Dutton & Co., Inc., New York, 1st edition, 1933. Translated from the Italian by Samual Putnam.