Semantic Technologies to Support the User-Centric Analysis of Activity Data

Mathieu d'Aquin, Salman Elahi and Enrico Motta

Knowledge Media Institute, The Open University, Milton Keynes, UK
{m.daquin, s.elahi, e.motta}@open.ac.uk

Abstract. There is currently a trend in giving access to users of online services to their own data. In this paper, we consider in particular the data which is generated from the interaction between a user and an organisation online: activity data as held in websites and Web applications logs. We show how we use semantic technologies including RDF integration of log data, SPARQL and lightweight ontology reasoning to aggregate, integrate and analyse activity data from a user-centric point of view.

1 Introduction

Social interactions on the Web, especially between individual users and organisations, rely on the exchange of personal data. As discussed in the article "Show Us the Data! (It is ours after all)" in the New York Times by Richard H. Thaler[1], while being heavily exploited by online organisations, these data are rarely made accessible to the users themselves. However, many initiatives have emerged recently arguing that obtaining and being able to exploit such data could be very beneficial to individual users. The *mydata* project in the UK[2] for example focuses on consumer data. At Google, the *Data Liberation Front*[3] has been formed to push the deployment of mechanisms allowing users to extract their data from Google services. In relation to this, there is currently a wide expansion of the idea of self-tracking, with new forms of applications being created based on social and personal data on the Web (see e.g., [1, 2]).

There are however specific challenges that appear when trying to apply such a user-centric perspective on activity data. Activity data typically sits in the logs of websites and Web applications, and are exploited by online organisations, in an aggregated form, to provide overviews of the interactions between the organisation's online presence and its users (most commonly in the form of website analytics). UCIAD[4] is a short project with the aim to experiment with the technological challenges that are faced when trying to invert the perspective

[1] http://www.nytimes.com/2011/04/24/business/24view.html
[2] http://www.bis.gov.uk/news/topstories/2011/Apr/better-choices-better-deals
[3] http://www.dataliberation.org/
[4] http://uciad.info

on activity data: provide individual users with an overview of their interactions with the online organisation.

This raises a number of challenges for which the use of semantic technologies seem to provide adequate solutions:

Fragmentation and heterogeneity: Activity data is typically held in log files that have different formats, and might not be easily integratable from one system (website, application) to another.

User identification: Recognising and identifying a user within the data is typically a problem faced by any activity data analysis. However, when taking a user-centric perspective, a user needs to be identified over several systems and the consequences of inaccurately recognising a user can be more critical.

Data analysis: Activity data is generally available through raw, uninterpreted logs from which meaningful information is hard to obtain.

Scale: Tracking user activities through logs can generate immense amounts of data. Typical systems cope with such scale through aggregating data based on clusters of users. Here, we need to keep the whole set of data for each individual user available to provide meaningful analysis of their interaction with the organisation in a user-centric way.

In this paper, we show how we investigated and handled these challenges through relying on semantic technologies, especially RDF for the low level integration and management of data, ontologies for the aggregation of heterogeneous data and their interpretation, and lightweight ontological reasoning to support customisable analysis of user-centric activity data. We also discuss how these components have been put together in a demonstrator platform, the UCIAD platform, providing user-centric views on activity data related to several websites of the Open University.

2 Activity Data Integration - Base Architecture

There are two reasons why we believe semantic technologies can benefit the analysis of activity data in general, and from a user-centric perspective in particular. First, ontology related technologies (including OWL, RDF and SPARQL) provide the necessary flexibility to enable the "lightweight" integration of data from different systems. Not only we can use ontologies as "pivot" models for data coming from different systems, but such models are also easily extensible to take account of the particularities of the systems available, but also to allow for custom extensions to deal with particular users, making personalised analysis of activity data feasible.

The overall architecture of the activity data infrastructure set up for the UCIAD project is shown in Figure 1. Its goal is to support the extraction from a variety of logs, of homogeneous representations of the traces of activity data present in these logs and store them in a common semantic store so that they can be accessed and queried by the user. We use RDF as a common data model, and a triple store providing SPARQL querying facilities for storing and accessing the

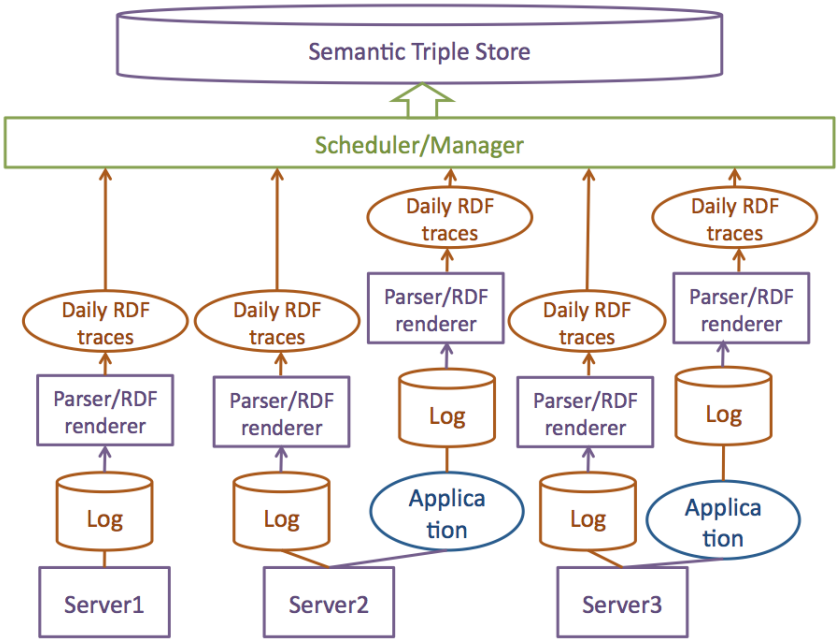

Fig. 1. Overview of the architecture of the UCIAD platform.

data.[5] Information from logs is extracted on a daily basis and represented using the ontologies described in the next section, which together with the semantic store represent the basis of the platform to provide user-centric views on activity data.

3 Aggregating Heterogeneous Activity Data - The UCIAD Ontologies

Compared to other domains, the advantage of user activities is that there is a lot of data to look at. This might be seen as an issue (from a technical and conceptual point of view), but in reality, this allows us to apply a bottom-up approach to building the ontologies necessary to achieve our goal: modelling through characterising the data, rather than through conceptualising the domain from established expert knowledge. It also gives us an insight into the scale of the task, and the need for adapted tools to support both the ontological definition of specific situations, and the ontology-based analysis of large amounts of traces of activity data.

[5] We use OWLIM (http://www.ontotext.com/owlim) which provides scalable storage and lightweight reasoning facilities.

3.1 Identifying Concepts and their Relations

The first step in building our ontologies is to identify the key concepts, i.e., the key notions, that we need to tackle, bearing in mind that our ultimate goal is to understand activities. We rely extensively on website logs as sources of activity data. In these cases, we can investigate requests both from human users and from robots automatically retrieving and crawling information from the websites. The server logs in question represent collections that can be seen as traces of activities that these users/robots are realising on websites. We therefore need to model these other aspects, which correspond to actions that are realised by actors on particular resources. We propose three ontologies to be used as the basis of the work in UCIAD:

The Actor Ontology is an ontology representing different types of actors (human users vs. robots), as well as the technical settings through which they realise online activities (computer and user agent).

The Sitemap Ontology is an ontology to represent the organisation of webpages in collections and websites, and which is extensible to represent different types of webpages and websites.

The Trace Ontology is an ontology to represent traces of activities, realised by particular agents on particular resources (here, mostly webpages). As we currently focus on HTTP server logs, this ontology contains specific sections related to traces as HTTP requests (e.g., HTTP methods are represented as instances of "Action" and HTTP response codes are included within "Response" type objects). It is however extensible to other types of traces, such as specific logs for particular applications.

3.2 Reusing Existing Ontology

When dealing with data and ontologies, reuse is generally seen as a good practice. Apart from saving time from not having to remodel things that have already been described elsewhere, it also helps anticipating on future needs for interoperability by choosing well established ontologies that are likely to have been employed elsewhere. We therefore investigated existing ontologies that could help us define the notions mentioned above. Here are the ontology we reused:

The FOAF ontology (http://xmlns.com/foaf/spec/) is commonly used to describe people, their connections with other people, but also their connections with documents. We use FOAF in the Actor Ontology for human users, and in the Sitemap Ontology for webpages (as documents).

The Time Ontology (http://www.w3.org/TR/owl-time/) is a common ontology for representing time and temporal intervals. We use it in the Trace Ontology.

The Action ontology (http://ontology.ihmc.us/Action.owl) defines different types of actions in a broad sense, and can be used as a basis for representing elements of the requests in the UCIAD Trace Ontology, but also as a base typology for actions. It itself relies on a number of other ontologies, including its own notion of actors.

While not currently used in our base ontologies, other ontologies can be considered at a later stage, for example to model specific types of activities. These include the Online Presence Ontology (OPO[6]), as well as the Semantically-Interlinked Online Communities ontology (SIOC[7]).

The current version of the ontologies developed as part of this work are available at https://github.com/uciad/UCIAD-Ontologies.

4 Identifying and Extracting User Activity Data

Once activity data have been extracted and represented according to the ontologies briefly described above, the next step consists in identifying and aggregating, in this data the traces of activities realised by a particular user, in order to create a user-centric view of his or her interactions with the considered systems (websites, applications).

4.1 Overview

The information the UCIAD platform collects regarding users can be seen as similar to the one basic analytics systems have. The user is rarely seen directly, as the interaction is mediated through a "user agent": a software programme running on a particular computer. Each HTTP request is associated with the ID of the user agent realising it, and the IP address of the corresponding computer. Several analytics systems use the combination of these two parameters to recognise a user with a reasonable level of accuracy. The disadvantage however is that the same user can be using different agents (e.g., different browsers) and different computers (or even mobile phones) to access the Web.

In UCIAD, we have the advantage that it is very likely that the user will connect to the UCIAD platform using the same agents and computers they usually use to access the Web, and especially the considered websites. The "settings" the user is using can therefore be detected at the time of logging in, and be attached to the user account. These settings will then be used to aggregate all the activity data that have been realised using the same computer and user-agent, and be added to the set of activity data for the particular user.

In addition, this provides a convenient mechanism to aggregate information realised on different computers and different settings. The user can log again in the UCIAD platform with a different browser, or a different device. When that happens, as described in Figure 2, the current setting will simply be added to the list of known settings for this user, and contribute another set of activity data around this particular user.

A setting, in our ontology, corresponds to a computer (generally identified by its IP address) and an agent (generally a browser), identified by a complex string such as

[6] http://online-presence.net/opo/spec/
[7] http://sioc-project.org/ontology

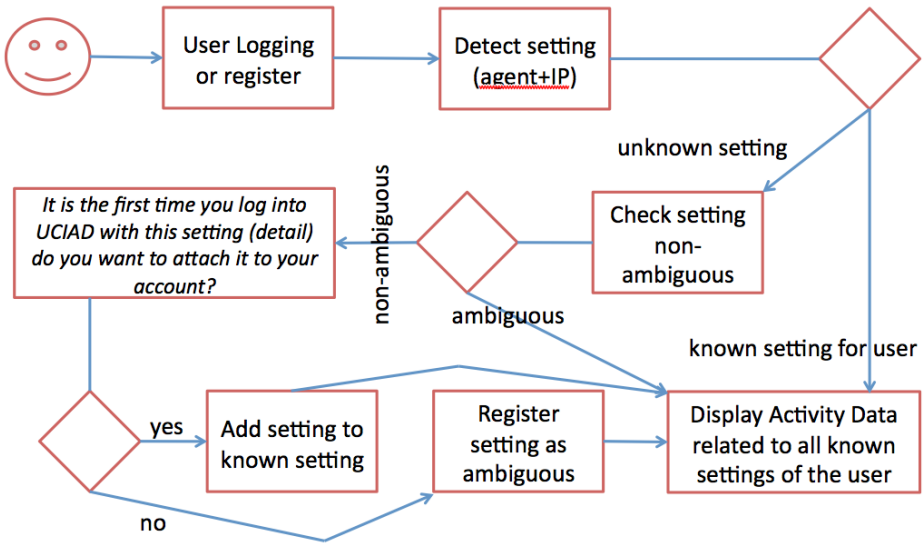

Fig. 2. Associating user accounts to their settings.

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_6) AppleWebKit/534.24
(KHTML, like Gecko) Chrome/11.0.696.68 Safari/534.24)
```

Such a setting can be associated to a user based on a representation following our ontologies described above, such as in the example below:

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:actor="http://uciad.info/ontology/actor/">
<rdf:Description rdf:about="http://uciad.info/actor/mathieu">
 <actor:knownSetting
  rdf:resource="http://uciad.info/actorsetting/4eafb6e074f46857b1c0b4b2ad0aa8e4"/>
 <actor:knownSetting
  rdf:resource="http://uciad.info/actorsetting/c97fc7faeadaf5cac0a28e86f4d723c9"/>
 <actor:knownSetting
  rdf:resource="http://uciad.info/actorsetting/eec3eed71319f9d0480ff065334a5f3a"/>
</rdf:Description>
<rdf:Description
 rdf:about="http://uciad.info/actorsetting/4eafb6e074f46857b1c0b4b2ad0aa8e4">
 <actor:hasComputer rdf:resource="http://uciad.info/computer/4eafb6e074f46857b1" />
 <actor:hasUserAgent rdf:resource="http://uciad.info/useragent/c0b4b2ad0aa8e4"/>
 </rdf:Description>
<rdf:Description rdf:about="http://uciad.info/computer/4eafb6e074f46857b1">
 <actor:hasIPAddress>187.108.24.45</actor:hasIPAdress>
```

```
</rdf:Description>
<rdf:Description rdf:about="http://uciad.info/useragent/c0b4b2ad0aa8e4">
 <actor:hasAgentID>Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_6)
   AppleWebKit/534.24 (KHTML, like Gecko) Chrome/11.0.696.68 Safari/534.24)
 </actor:hasAgentID>
</rdf:Description>
</rdf:RDF>
```

This indicates that the user http://uciad.info/actor/mathieu has three settings. These settings are all on the same computer and correspond to the Safari and Chrome browsers, as well as the Apple PubSub agent (used in retrieving RSS feeds amongst other things).

4.2 Extracting User-Related Data

Managing the activity data regarding a particular user corresponds to creating a sub-graph of the complete graph of raw activity data we collect from logs, based on the information about the known settings of the user. To identify a user, we rely here on the settings used to realise the activity. Each trace of activity is realised through a setting (linked to the trace by the *hasSetting* ontology property). Knowing the settings of a user therefore allows us to list the traces that correspond to this particular user through a simple query. Using a SPARQL Construct query, we can create a model, i.e. an RDF graph, that contains all the information related to the user's activity on the considered websites:

```
PREFIX tr:<http://uciad.info/ontology/trace/>
PREFIX actor:<http://uciad.info/ontology/actor/>
construct {
  ?trace ?p ?x.
  ?x ?p2 ?x2.
  ?x2 ?p3 ?x3.
  ?x3 ?p4 ?x4
} where{
  <http://uciad.info/actor/mathieu> actor:knownSetting ?set.
  ?trace tr:hasSetting ?set.
  ?trace ?p ?x.
  OPTIONAL {{?x ?p2 ?x2}.
  OPTIONAL {{?x2 ?p3 ?x3}.
  OPTIONAL {?x3 ?p4 ?x4}}}
}
```

The results of this query correspond to all the traces of activities in the collected data that have been realised through known settings of the user http://uciad.info/actor/mathieu, as well as the surrounding information. These data, materialised as an RDF graph, can therefore be considered on its own, as a user-centric view on the activity data available through integrated logs.

4.3 Managing Access Right over Semantic Data

We store, manipulate and reason over activity data using Semantic Web technologies, namely RDF, a triple store with inference capabilities and SPARQL for querying. As part of the UCIAD platform, we needed a mechanism to restrict the queries being sent to only the part of the data that the current user has access to: his/her own subgraph of activity data.

Unfortunately, most current triple stores, and especially the one we are employing, do not provide sufficiently fine-grained access control mechanisms, allowing to associate sub-graphs to particular users. We therefore implemented our own mechanism, which can be seen as a generic recipe for access control over activity data.

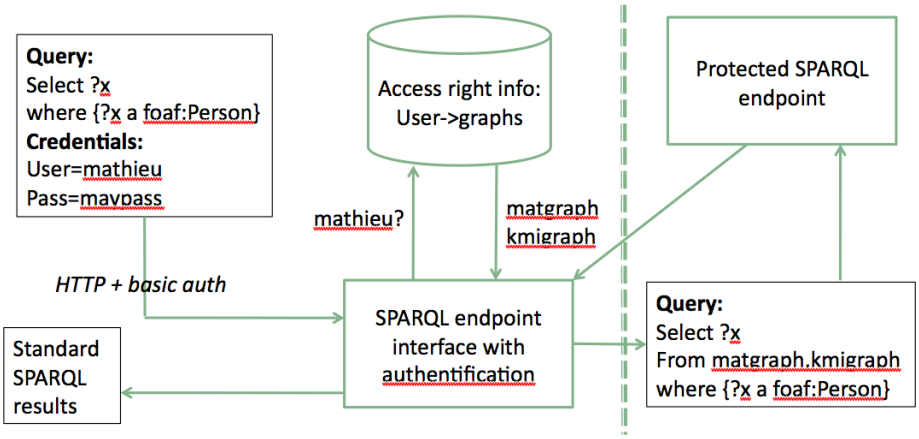

Fig. 3. Overview of the mechanism for access right to data in a SPARQL endpoint.

The idea, as depicted in Figure 3, is that the actual SPARQL endpoint giving access to all the data for all the users is being hidden using standard security measures so that it can only be accessed by our own system. We then implement a "proxy SPARQL endpoint" that can handle basic HTTP authentication. When receiving a query, this proxy endpoint will check the credential of the user and see what sub-graphs the user has access to, so that it can modify the query to restrict it to these sub-graphs only (using the FROM clause in SPARQL). It can then send the query to the real, hidden SPARQL endpoint and forward the results back to the user.

While this mechanism is relatively simple, it offers an appropriate level of flexibility, allowing to define arbitrary subgraphs and user definitions as a model for access control.

5 Interpreting and Analysing Activity Data through Lightweight Ontology Reasoning

Here, we want to use the ontologies we have created, and extend them, so that they can support the interpretation and analysis of the extracted activity data. What we want to achieve is, through providing ontological definitions of different types of activities and resources, to be able to characterise different types of traces and classify them as evidences of particular activities happening.

The first step in realising such inferences is to characterise the resources over which activities are realised – in our case, websites and webpages. Our ontologies define a webpage as a document that can be part of a webpage collection, and a website as a particular type of webpage collection. As part of setting up the UCIAD platform, we declare in the RDF model the different collections and websites that are present on the considered server, as well as the URL patterns that make it possible to recognise webpages as parts of these websites and collections. These URL patterns are expressed as regular expressions and an automatic process is applied to declare triples of the form $page_1\ isPartOf\ website_1$ or $page_2\ isPartOf\ collection_1$ when the URLs of $page_1$ and $page_2$ match the patterns of $website_1$ and $collection_1$ respectively.

The base ontologies we have defined can then be extended to represent particular categories of resources, depending on their properties. We for example declare a particular website as a $Wiki$. We can also declare a webpage collection that corresponds to RSS feeds, using a particular URL pattern, and use an ontology expression to declare the class of $WikiUpdate$ as the set of webpages which are both part of a $Wiki$ and part of the $RSSFeed$ collection, i.e., in the OWL abstract syntax

```
Class(WikiUpdateFeed complete
    intersectionOf(Webpage
      restriction(isPartOf someValuesFrom(RSSFeed))
      restriction(isPartOf someValuesFrom(Wiki))))
```

We can similarly define the activity of checking and federating updates from a wiki by creating the class of traces of activities (requests) realised on a $WikiUpdateFeed$ using an $RSSClient$ as user agent. Another example would be defining the class $ExecutingASPARQLQuery$ as the one of sending a request to a page of the type $SPARQLEndpoint$ using a *query* parameter.

Such definitions can be added to the repository, which, using its inference capability, will derive that certain pages are $WikiUpdateFeed$s, and certain activities correspond to $ExecutingASPARQLQuery$ without this information being directly provided in the data, or the rule to derive it being hard-coded in the system. We can therefore engage in an incremental construction of an ontology characterising websites and activities generally, in the context of a particular system, or in the context of a particular user.

6 Implementation: the UCIAD Platform

We realised the UCIAD platform as a demonstrator, where a user can register to the platform with some setting details and browse his or her activity data as they appear on several Open University websites (mostly, an internal wiki system and the Open University's linked data platform – data.open.ac.uk).[8]

The current "in development" version of the platform implements and demonstrates the following components described above:

User management: As the user registers into the UCIAD platform, his current setting is automatically detected, and other settings (other browsers) that are likely to be associated to him or her are also included. As the user registers, the settings are associated to his account and the activity data realised through these settings are extracted.

Extracting user-centric activity data: As described in Section 4.2, the settings associated with the user are used to extract the activity data around this particular user, creating a sub-graph corresponding to his or her activity.

Ontologies to make sense of activity data: The ontologies are used in structuring the data according to a common schema and to provide a base to homogeneously query data coming from different systems. As discussed above, they can also be extended (specified) so that different categories of activities and resources can be represented, and reasoned upon.

Ontological reasoning for analysis: Activity data is organised according to different categories (traces, webpages, websites, settings, etc.) coming from the base ontologies, but also according to classes of activities, resources, etc. that have been specially added to cover the websites and the particular user in this case (see Section 5). Here, we extended the ontologies in order to include definitions of activities relevant to the use of a wiki and a data platform. For example, we define "Executing a SPARQL Query" as an activity that takes place on a SPARQL endpoint with a "query" parameter, or "Checking Wiki Updates" as an activity on a Wiki page that is realised through an RSS client.

Browsing data according to ontologies: We rely on an homemade "browser" that we use in a number of projects and that can inspect ontology classes and members of these classes, generating graphs and simple statistics for these classes and members.

7 Discussion and Future Work

While the UCIAD platform provides an interesting first attempt at demonstrating the feasibility of user-centric activity data based on semantic technologies, a number of challenges are left to be considered before such technologies could be deployed in realistic settings to provide Web users with an appropriate view on

[8] see http://uciad.info/ub/2011/08/final-post-putting-things-together-with-a-demo/ for a description and a video of this demonstrator.

their own activity data.

The first, technical challenge is scalability. Indeed, triple stores such as OWLIM can now handle very large amounts of data (see the benchmark tests in [3, 4]). However, activity data in the form of traces from logs are enormous. Indeed, an average Web server from the Open University would serve a few million requests per month. Each request (summarised in one line in the logs) is associated with a number of different pieces of information that re-factor in terms of our ontologies, concerning the actor (IP, agent), the resource (URL, website it is attached to, server), the response (code, size) and other elements (time, referrer). We can obtain between 20 and 50 triples per request. This leads us to amounts of data in the order of 100 million triples per month per server (each server can host many websites). In theory, OWLIM should cope with such a scale, even if we consider several servers over several months. However, the data we are uploading to OWLIM is complex, and has a refined structure. Some objects (user settings, URLs) would appear very connected, while others would only appear in one request, and share only a few connections. From our experience, it is not only the number of triples that should be considered, but also the number of objects. A graph where each object is only associated with 1 other object through 1 triple might be a lot more difficult to process than one with as many triples, but shared amongst significantly less nodes (see [5]).

This scale issue is amplified when inference mechanisms are applied. OWLIM handles inferences at loading times. This means that not only the number of triples uploaded onto the store are multiplied through inferences, but also that immensely more resources are required at the time of loading these triples, depending not only on the size of what is uploaded, but also on its complexity (and, as mentioned above, our data is complex) and on the size of what is already stored. Originally, our approach was to have one store holding everything with inferences, and to extract from this store data for each user. We changed this approach to one were the store that keeps the entire dataset extracted from logs does not make use of inference mechanisms. Data extracted for each user are then transferred into another (necessarily smaller) store for which inferences apply.

A less technical challenge for approaches to activity data relying on a user-centric perspective is the identification of user-related data and their distribution. Indeed, as we explained in Section 4, we identify users based on a number of indicators detected at the time the user is registering and logging in. These indicators are far from being 100% accurate. Other types of systems can cope with inaccuracy as they are generally eliminated or reduced when the data is being aggregated. However, here, providing activity data to the wrong user could create critical privacy issues that need to be considered. More robust security mechanisms, as well as more accurate user identification mechanisms (using for example the cookies employed by Web tracking systems) would need to be deployed.

Another crucial element concerns the distribution of the data. One of the important aspects of user-centric data is that the user should be able to export his or her own data, in order to exploit them for their own benefit. The ownership of the data is not however very clear in this case. It is data collected and delivered by our systems, but that are produced out of the activities of the user. We believe that in this case, a particular type of license is needed, which would give control to the user on the distribution of their own data, but without opening it completely.

References

1. d'Aquin, M., Rowe, M., Motta, E.: Self-tracking on the web: Why and how. In: W3C Workshop on Web Tracking and User Privacy. (2011)
2. d'Aquin, M., Elahi, S., Motta, E.: Personal monitoring of web information exchange: Towards web lifelogging. In: Web Science 2010, WebSci10: Extending the Frontiers of Society On-Line, poster. (2010)
3. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. Journal of Web Semantics **3**(2) (2005)
4. Bizer, C., Schultz, A.: The berlin sparql benchmark. International Journal on Semantic Web and Information Systems **5**(2) (2009)
5. Duan, S., Kementsietsidis, A., Srinivas, K., Udrea, O.: Apples and oranges: A comparison of RDF benchmarks and real RDF datasets. In: SIGMOD. (2011)