

# Semantic Monitoring of Personal Web Activity to Support the Management of Trust and Privacy

Mathieu d'Aquin, Salman Elahi, Enrico Motta

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

**Abstract.** For individual Web users, understanding and controlling their exchange of personal data is a very complex task as they interact, sometimes unknowingly, with hundreds of different websites. In this paper, we present a set of tools and an experiment dedicated to monitoring a user's Web activity in order to build an observed model of his behavior in terms of the trust given to accessed websites and of the criticality of the data exchanged. By exposing such a model to the users and allowing them to interact with it, we provide ways for users to be better informed about their own behavior with respect to privacy, and ultimately, to better control their own data exchange.

## 1 Introduction

Web users send data of varying degrees of criticality, to websites which they trust to various levels. Reasonable users would for example agree for a well known online retailer website to know about their address, while would normally not be conformable with sending data more critical than their screen resolution to a completely unknown or untrusted website. Indeed, it is expected that an informed and rational user naturally implements a “personal policy” relying on an implicit model of the trust relationship they have with websites, and of the criticality of their own data. However, the inherent complexity, the fragmentation and the implicitness of Web data exchange makes it almost impossible for any user to be adequately informed. Not many users would be able to list the websites to which they have sent a particular information such as their e-mail address for example. Even more difficult is to know how much information is transferred to an unknown website, as a side effect of accessing a trusted one.

In this paper, we present a tool and an experiment intended to demonstrate how we can derive *implicit models* of trust in domains (i.e., websites) and criticality of data from locally generated traces of the user's activity on the Web. We rely on data generated through a logging mechanism that keeps track in RDF of any communication through the HTTP protocol occurring on the user's computer [1]. We then develop simple models and tools to extract and represent data transfers from the generated logs, and to map these data transfers onto a semantic profile of the user.

The main contribution of this paper is to show how we can derive from this data notions of the *observed trust in domains* and *observed criticality of data*. Indeed, intuitively, these two notions relate with each other in the sense that we consider that a website is highly trusted if it has been sent very critical data, while a piece of data is not critical if it was sent to many untrusted websites. Making such notions explicit to the user and allowing them to explore the underlying data can be very useful as it makes emerge implicit relationships between the user, websites and his data, which might not be expected or intended by the user. In addition, we propose an interactive tool allowing users to ‘align’ the observed behavior with their ‘intended’ model of trust and criticality, in order to identify conflicts that can be acted upon.

As a concrete experiment for these tools, we detail at each step the results obtained using the Web activity logs generated by the first author of the paper over a period of 2.5 months.

## 2 Related Work

As lengthly described in [2], trust is a central element of any social interaction, and therefore, of any exchange on the Web. Indeed, beyond the Web 2.0 emphasis on the Web as a social platform, where people can exchange and share information, experience and more, any communication on the Web appears to be a social interaction between a person (Web user) and a website, which ultimately represents another person, group of people or organization. Therefore, a lot of attention has been dedicated to the notion of trust in the research community on the Web [3, 4]. While it would be out of the scope of this paper to detail these initiatives, we can mention as examples works where trust is considered a value attached to information, or to the provider of information, and that quantify the confidence one has that the information provided is correct (see e.g., [5]). In other cases, trust relates more to the notion of privacy, where it is attached to the recipient of some (usually critical, personal) data and corresponds to the confidence one has that the data would not be used for unintended purposes. For example, the *Platform for Privacy Preferences*<sup>1</sup> (P3p) provides a framework for websites to express their privacy practices, declaring explicitly in which way they can be trusted in handling user data. The work presented here is more directly related to this second category. However, contrary to P3P which takes a ‘website-centric’ perspective on trust, we consider here a user-centric view on trust in websites (domains) and on the criticality of the data sent to these websites. The intent is to derive from the traces of the user’s activity a model of his own trust relationship with the various websites he interacts with.

A range of tools exist already to support a user in monitoring his own Web activity, including tools used to debug communication protocols. More related to our approach here, we can mention for example Google Web History<sup>2</sup> and

---

<sup>1</sup> <http://www.w3.org/P3P/>

<sup>2</sup> <https://www.google.com/history/>

the Attention Recorder<sup>3</sup>. Both take the form of local applications (e.g., a plugin for popular web browsers), and record accesses to websites in order to build a record of Web activities. However, such tools are still limited in the sense that they record only a restricted amount of information (websites explicitly accessed through the Web browser) and only allow usage of the data which is directly intended by the tool (provide reports and improve the results of search in one case, sharing ‘attention data’ in the other). As it appears in our experiment (see below), data exchange on the Web is a very complex activity, often fragmented and partly implicit. Dedicated tools are therefore needed to monitor them and derive equally complex trust relationships between the user and the corresponding websites.

### 3 Tracking User Web Activities for Data Transfer

Our goal in this paper is to use the traces of users’ Web activity to build a model of their trust relationship with websites, and of the criticality of their own data. We first need to briefly introduce the underlying technology allowing us to obtain complete RDF models of Web activity and data transfer. We also detail at each step the results obtained in our experiment realized over a period of 2.5 months with the Web activity of the first author of this paper.

#### 3.1 Logging Web Activity

In order to represent a sufficiently broad overview of personal data transfer on the Web, we need a tool which would fulfill two main requirements: 1- it needs to be transparent to the user, acting in background without disrupting normal Web activities; and 2- it needs to collect information as complete as possible, in particular, independently from the Web agent used (various Web browsers, but also many other tools such as online music programs—e.g., iTunes<sup>4</sup> and spotify<sup>5</sup>, e-mail clients—getting Web images and other content from inside e-mails, etc.) For these reasons, we implemented our logging mechanism as a Web proxy, running on the local computer of the user. A proxy is a tool that acts as an intermediary between a client and external servers the client is trying to connect to. Web proxies are often used in organizations to implement cache control mechanisms for all the Web users inside the organization’s network.

Here however, we use a Web proxy locally. Web communications can be redirected to it through the system preferences so that any HTTP request going out of the user’s computer (and any response back) is intercepted, logged and re-directed to the right destination (which could be another Web proxy). As shown in Figure 1, the logs of the Web activity collected through this tool are

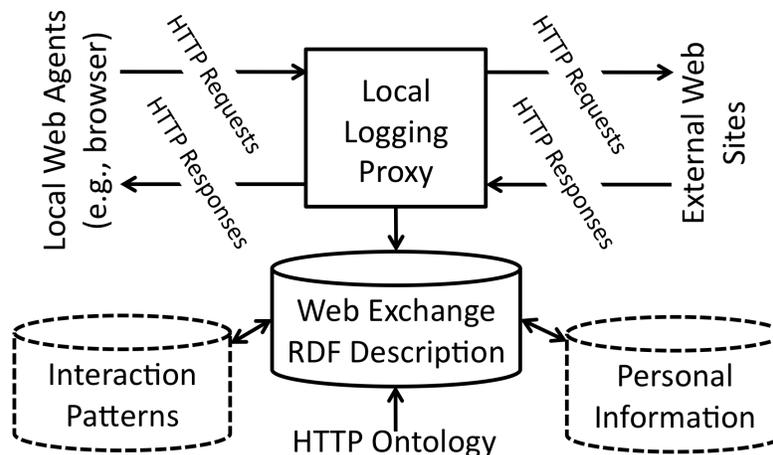
---

<sup>3</sup> <https://addons.mozilla.org/en-US/firefox/addon/3569>

<sup>4</sup> <http://www.apple.com/itunes/>

<sup>5</sup> <http://spotify.com>

represented in RDF, using a simple, ad-hoc HTTP ontology<sup>6</sup>. These logs record the complete information included as part of the HTTP protocol (e.g., destination, agent, cache information, referrers, etc.), as well as pointers to the actual data exchanged, which is saved on the local file system.



**Fig. 1.** Overview of the Web activity logging system.

In our experiment, this tool has recorded over 3 million HTTP requests during a period of 2.5 months, spanning over many different Web agents on the user’s local computer, and representing all together 100 million RDF triples and 9GB of data (in the RDF/XML syntax). The scalability of the tool and its ability to process such data in real time represents a major challenge for this work. This is however outside the scope of this paper and will be treated as future work.

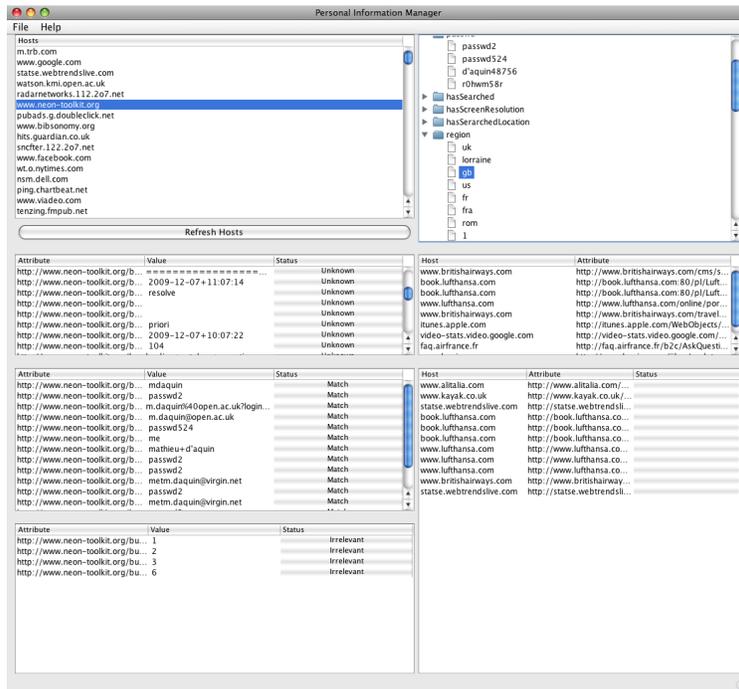
### 3.2 Investigating Data Transfer

Of course, the data collected using the tool described above contains a lot more information than necessary for the purpose of investigating data transfer to model trust in websites and criticality of data. We therefore extract from this data a subset that corresponds to elements of data that are being sent by the user’s Web agents to external websites. We use a simple SPARQL query to obtain the list of requests to which data was attached. This includes HTTP GET requests with parameters (e.g., in <http://www.google.co.uk/search?q=keywords>

<sup>6</sup> This ontology was built for the purpose of the tool, to fit the data, but can be seen as an extension of <http://www.w3.org/TR/HTTP-in-RDF/>

the parameter is  $q=keyword$ ), as well as HTTP POST requests where the same kind of parameters are enclosed in the content (data) part of the request. Parsing these parameters in both the URLs of GET requests and the data content of POST requests, we build a smaller dataset made of triples of the form  $\langle website, hasReceivedWithParam-P, v \rangle$ , where  $website$  is the host to which the request was addressed,  $P$  is the attribute part of a parameter, and  $v$  is the value part.

Based on the activity logs in our experiment, we extracted more than 33,000 of such triples.



**Fig. 2.** Screenshot of the Data Transfer Log to User Profile mapping tool. On the right hand side is the data transfer log, showing different websites and the data they received. On the left hand side is the profile created from mapping this log (top), as well as suggestions for additional mappings (bottom).

### 3.3 Mapping to Personal Data

While the data extracted above only concerns data sent from the user to external websites, not all of it relates to personal information and it is not in this form easily interpretable as such. In order to extract from such a data transfer log *relevant data*, we built a tool that allows the user to easily identify personal information in it. Without going into the details (this tool is described

in [6]), it provides mechanisms for the user to create mappings between the parameters used in particular websites (e.g., `http://qdos.com/signin#username`, `http://spreadsheets.google.com/ccc #email`) and attributes of a very simple model of the user profile (e.g., `UserName`, `e-mail`). The attributes in the profile might initially exist or might be created on demand, as required by the creation of a mapping. Once a mapping is created, the role of the tool is first to use it to populate the user profile with values from the data (e.g., `e-mail=m.daquin@open.ac.uk`). It also suggests additional mappings by looking, in the data transfer log, at where values already added to the profile appear.

From the data extracted for our experiment and using the interactive interface described above (see screenshot Figure 2), we built a profile made of 36 attributes and 598 values, creating 1,113 mappings to 184 different websites. Re-integrating such information into the RDF log data could allow for many different ways of studying and analyzing the user behavior [1]. Here, we focus on deriving from it models of observed trust in websites and data criticality.

## 4 Observed Trust in Websites and Criticality of Data

The data obtained from the tool above contains information about both the user profile and, through the mappings, about the websites to which each piece of information has been sent. This constitutes the basis of our model of trust in websites and criticality of data. First, we introduce some definitions concerning websites, domains and data pieces.

### 4.1 Basic Notions

We identify websites through their second level *domain* (SLD<sup>7</sup>) names (e.g., `google.com`, `snCF.fr`). The list of domains is automatically extracted from the data obtained in the previous step, using the URLs to which data was sent through HTTP requests. We call  $D$  the set of all the domains  $d_i$  in our dataset. In our experiment, there were 123 different domains that received data from the user profile.

We consider the notion of *data piece* to represent an element of information from the user profile, which was sent to one or more domains. Here, we use the attributes of the profile to correspond to data pieces (e.g., `passwd` or `e-mail` are data pieces). We call  $P$  the set of data pieces  $p_i$  present in our dataset. In our experiment, there were as many data pieces as attributes in the profile (i.e., 36).

Finally, we define two simple functions, to represent the list of data pieces received by a particular domain, and the list of domains to which a piece of data was sent, i.e.,

- $R(d_i) \subseteq P$  represents the set of data pieces received by the domain  $d_i$
- $S(p_i) \subseteq D$  represents the set of domains to which the piece of data  $p_i$  was sent

---

<sup>7</sup> [http://en.wikipedia.org/wiki/Second-level\\_domain](http://en.wikipedia.org/wiki/Second-level_domain)

For example, in our experiment,  $R(\text{lip6.fr}) = \{\text{username, passwd, hasReviewed}\}$  and  $S(\text{city}) = \{2o7.net, britishairways.com, ter-sncf.com, google.com\}$ , all this information being extracted directly from the Web activity logs and the mappings to the user profile.

## 4.2 Computing the Observed Trust in Domains and Data Criticality

Our goal here is, relying on the simple notion of data transfer defined above, to analyse the behavior of the user and derive what is expected to be his implicit trust relationship with the considered websites, and the correlated levels of criticality he seems to associate to each of the considered pieces of data. Crucially, these two notions are highly inter-dependent. Indeed, on the one hand, it is natural for an external observer to assess the trust somebody has in another agent based on the information he is prepared to disclose to this external agent. For example, if I consider my mobile phone number as a critical information, and disclose it to a particular website, this seems to indicate a high level of trust in this particular website. On the other hand, assessing the criticality of a piece of data can be done by considering how much this information is disclosed to external, varyingly trusted agents. The information about my screen resolution for instance might not be considered very critical, since I have provided it to many different website, most of them not very trusted.

On the basis of these simple intuitions, we define two functions,  $T(d_i) \in [0..1]$  and  $C(p_i) \in [0..1]$ , representing the levels of observed trust in a domain  $d_i$  and of criticality of a piece of data  $p_i$  respectively. These measures are dependent on each other according to the two equations (1) and (2) below:

$$T(d_i) = \max_{p_j \in R(d_i)} (C(p_j)) \quad (1)$$

$$C(p_i) = \frac{1}{1 + \sum_{d_j \in S(p_i)} 1 - T(d_j)} \quad (2)$$

Intuitively, (1) translates the idea that the level of trust associated with a domain  $d_i$  corresponds to the level of criticality of the most critical piece of data  $d_i$  has received. Equation (2) is slightly more complex. It is meant to give a high value of criticality to a piece of data  $p_i$  if  $p_i$  was sent only to a small number of highly trusted domains, and a low value if it has been sent to a high number of not trusted domains.

The most obvious problem with these measures is of course their interdependence. In practice, we consider them as sequences with the values of criticality  $C(p_i)$  for each  $p_i \in P$  at a time  $t$  calculated on the basis of the values of trust  $T(d_j)$  for domains  $d_j \in D$  at the time  $t - 1$ . Using initial values of 0.5 for both trust and criticality, these measures converge to a stable state (with a precision of 0.0001) in 285 iterations on our dataset. The result is that each domain and each piece of data is associated with a level of observed trust and criticality respectively, which the user can then inspect to check to which extent it corresponds to his own intended, implicit model of trust and criticality. An interactive, visual tool to support the user in such a task is presented in the next section.



(in our case, domain trust and data criticality). This representation shows each object as a ‘bubble’ along an axis (representing the score), with the size of each bubble also representing the score. Applied on our data, Figure 3 shows in the top half the visualization of the computed trust in domains (purple bubbles) and in the bottom half the computed criticality for the considered pieces of data (orange bubbles). While relatively simple, this visualization allows the user to quickly identify for example which are the most trusted domains and what is the relation between the values of criticality for different pieces of data (e.g., `first name` is less critical than `full name`, or `e-mail` is half as critical as `postal address`).

## 5.2 Exploring the Data

In addition to providing a simple representation for the measures constituting our model of observed trust in domains and data criticality, it is important to provide to the user ways to interact with the data, so that he can explore further the relation between websites’ domains, the data they received, trust and criticality. Indeed, one important information hidden in our model concerns what information has been sent to which domains. Here, by selecting in the top panel the bubble corresponding to a given domain, the list of pieces of data it has received is highlighted in the bottom panel. In our example Figure 3, the domain `elsevier.com` was selected, showing that this domain has received data of very varying levels of criticality. In the same way by selecting a piece of data in the bottom panel, the domains which have received this particular piece of data would be highlighted.

Such a simple way to explore the collected data is crucial to our approach to managing trust and privacy. Indeed, it allows the user to answer questions in a way that would not be possible without the appropriate monitoring of Web traffic, such as “Which websites know my e-mail address?” or “What does `google-analytics.com` know about me?”. In a more abstract way, it also allows the user to explore his own behavior, by showing him the criticality of the data sent to particular websites, and what it says about the trust he, in appearance, is giving to them.

## 5.3 Interacting with the Model and Detecting Conflicts

One of the most important advantage of exposing the observed behavior of the user with respect to trust in domains and data criticality is that it gives him the ability to compare it to his intended behavior. In other terms, the user should be given the ability to disagree with the model, to try to correct it, and to detect fundamental conflicts between his own view and what the model can derive from the observed behavior. Indeed, it appears obvious that the computed model sometimes comes up with values which are fundamentally different from what the user would have expected, considering for example the information about his e-mail address as being not very critical and associating high values of trust to websites with relatively unclear privacy policies (e.g., `lip6.fr`).

To support the user in expressing this mismatch between his intended behavior and the observed model, our tool allows him to manually set the values of the trust in a domain or criticality of a piece of data. In practice, he can drag one of the bubbles into another place along the axis, fixing the value for the considered measure. While a manually set measure will not anymore be affected by the model described in Section 4, it will continue to impact on the trust and criticality of other, not modified domains and pieces of data. In other terms, as the user moves a domain or a piece of data around, the influence of this modification on the model can be directly seen through other bubbles moving in the same direction. For example, reducing the trust value for `elsevier.com` would directly affect the criticality associated with the pieces of data `elsevier.com` has received, and indirectly, the trust that is associated with other domains. As the visualization is updated dynamically, this is translated into a number of bubbles moving in both panels, following the movement of the one being selected and set manually. Interestingly, while this provides a way to interact with the model and understand the relation between trust and criticality, it can also be used to identify interesting correlations resulting in simultaneous movements, derived from indirect calculations. Indeed, moving for example the bubble corresponding to `last-name` in the bottom panel not only makes domain bubbles to move accordingly in the top panel, but also results in the `first-name` data piece being updated, showing a strong relationship between these two pieces of data and their exchange with various websites.

Using the mechanisms described above, the user, starting from the computed model translating the observed behavior, can build his intended model based his own view on trust and criticality. However, manually setting values for trust and criticality inevitably results in conflicts between this intended, declared model and the fundamental assumption underlying the computed model: That untrusted websites should not receive critical data. We therefore define and detect a conflict as a significant positive difference between the manually set value for the criticality of a piece of data and the manually set value for trust in a domain to which this piece of data was sent. More formally, we define the set  $C$  of conflicts in a user-corrected model as  $C = \{(d_i, p_j) | d_i \in MD \wedge p_j \in MP \wedge p_j \in R(d_i) \wedge C_m(p_j) - T_m(d_i) > \epsilon\}$ , where  $MD$  is the set of domains for which the trust is manually set,  $MP$  the set of pieces of data for which the criticality is manually set,  $C_m(p_j)$  is the manually set value of criticality for a piece of data  $p_j$ ,  $T_m(d_i)$  is the manually set value of trust for the domain  $d_i$  and  $\epsilon$  is a given constant above which the level of conflict is considered significant. Figure 4 shows two examples of detected conflicts, as they appear to the user and ranked according to the value of the difference  $C_m(p_j) - T_m(d_i)$ . In these examples, the user has indicated that the domain `elsevier.com` was less trusted than observed, and that the pieces of data `hasReviewed` (journal articles that were reviewed by the user) and `e-mail` were more critical then computed by the model, resulting in the displayed conflicts.



Fig. 4. Examples of conflicts detected after the user had modified the model.

## 6 Going Further: Semantically Enriching Traces of Web Activity for Personal Privacy Policies

As shown in the previous sections, tools keeping track of the user’s Web activities can make emerge models of the observed trust in websites and criticality of the data. Exposing these models to the user and allowing him to interact with them provides the user with a way to better control his own privacy, by informing him of possibly unintended behaviors (i.e., conflicts) so that he can act on them. Indeed, in our experiment, many of such conflicts arise and elements that emerged from the data appeared sometimes very surprising (e.g., the amount of critical information being unknowingly sent to `google-analytics.com` as a result of other websites using this tool).

In building the above described techniques, the use of semantic technologies appears very useful, in order to provide flexible models of Web activity logs, which are mapped onto users’ personal information. In addition, it provides us with the possibility to enrich the produced models with external data, to interlink it, so that richer ways to explore the user’s own activity logs and trust models can be used. One simple example would be to integrate for each domain the corresponding ‘semantified’ information from its registrar (i.e., using the *whois* utility). Such information describes the people/companies who own the domain with contact information and addresses. By relating it to geo-localization data (e.g., the geonames dataset<sup>8</sup>), pieces of data could be reconnected not only to where they were sent on the Web, but could also to more concrete elements, allowing for example to explore the implications in terms of the privacy laws applying to different Web interactions.

One of the obvious next steps for this work is to provide the user with support not only to understand his own behavior, but also to directly act on it. It would be for example easy to adapt the logging tool described in Section 3.1 to implement user-defined rules instructing the tool to reject any attempt to send data to particularly untrusted website or to alert the user before sending data of certain levels of criticality to unknown websites. Here as well, the use of semantic technologies would appear crucial in order that these rules are defined with the appropriate levels of expressivity and granularity. This would allow for example a user to declare that no data above a certain level of criticality should be sent

<sup>8</sup> <http://geonames.org>

to any analytics website, while defining clearly the conditions for a domain to be classified as an analytics website and exploiting information pulled from, e.g., DBPedia<sup>9</sup> to enrich the information known about a given domain so that it is sufficient to test these conditions.

## 7 Conclusion

In this paper, we showed how tracking the data exchanges for an individual user can help in defining personal models of trust in websites and data criticality, providing the first elements of a platform to better monitor and control the exchange of personal information on the Web. In the short term, the realization of such a platform raises many technical challenges, including the ability to process such amounts of data in real time, without impacting on the user's Web experience. In the longer term, many refinements are currently being investigated to provide more support to the user in exploring and controlling their own data exchange, which implies relating the user data with external, semantic data. Another interesting further step consists in integrating a social aspect to the management of personal privacy policies, allowing users to share their experience with particular domains. This can include sharing trust information concerning particular websites, but also making the mappings between websites' attributes and personal information available for others to reuse.

In addition, this work on using Web activity logs to better understand the behavior of Web users with regards to privacy and trust can be seen as considering one aspect of a broader domain of study, observing the full range of activities on the Web to derive useful models, both for the users themselves, but also for researchers in technological and non-technological areas. As such, it appears important to open similar experiments to the one described in this paper to larger groups of users, preferably with different backgrounds, interests and uses of the Web.

## References

1. d'Aquin, M., Elahi, S., Motta, E.: Personal monitoring of web information exchange: Towards web lifelogging. In: Web Science Conference (poster presentation, to appear). (2010)
2. O'Hara, K.: Trust – from Socrates to Spin. Icon Books (2004)
3. Hoffman, R.R., Lee, J.D., Woods, D.D., Shadbolt, N., Miller, J., Bradshaw, J.M.: The dynamics of trust in cyberdomains. *IEEE Intelligent Systems* **24** (2009) 5–11
4. Golbeck, J.A.: Computing and applying trust in web-based social networks. PhD thesis, College Park, MD, USA (2005) Chair-Hendler, James.
5. Rowe, M., Butters, J.: Assessing trust: Contextual accountability. In: First Workshop on Trust and Privacy on the Social and Semantic Web. (2009)
6. Salman Elahi, M.d., Motta, E.: Who want a piece of me? reconstructing a user profile from personal web activity logs. In: International ESWC Workshop on Linking of User Profiles and Applications in the Social Semantic Web. (2010)

---

<sup>9</sup> <http://dbpedia.org>