Personalizing web surfing with semantically enriched personal profiles

Anupriya Ankolekar, Denny Vrandečić

Institute AIFB, University of Karlsruhe, Germany
{anupriya,denny}@aifb.uni-karlsruhe.de

Abstract. Personalization mechanisms on the web today are clumsy and obtrusive, because users need to log in to multiple websites and enter their personal information and preferences separately for each. In addition, the user profile is different for each website and cannot be combined with other information on the web. Using Semantic Web technologies, in particular FOAF, we can identify the person browsing to the website. In this paper, we propose an extension of the HTTP `GET` method to include a new parameter that points to the URL of the user's FOAF file. This simple, but powerful extension enables the web server to use information contained in the person's FOAF file to personalize the web pages returned. We also present a proof-of-concept implementation by customizing our institute welcome page using a visitor's FOAF file.

1 Introduction

No one knows you are a dog on the Internet, or so the saying goes. While this may have its advantages, letting websites know who you are enables them to personalize your web surfing experience, serving you information customized for your needs and preferences. Knowing the interests, activities, acquaintances or accessibility problems of a user, the web site can adapt the web pages it serves and offer a personalized, familiar and welcoming experience for the user. The personalizations may include content personalization (show me the things I am interested in), or link personalization (show me only those links that I am likely to click on) and structure/navigation personalization (reorder the website as I am likely to understand and search through it). Several studies have explored the use of personalization and knowledge of people's activities on a website to support web browsing and have found it to considerably enhance the browsing experience for users [15].

A key requirement for enabling such personalization is that the web server must be able to identify the person visiting the website and know her characteristics and preferences. Currently, the vast majority of sites attempt to personalize the user's browsing experience by requiring the user to create an account on the website and login every time a personalized service is used. *Amazon* and *Yahoo!*, for example, follow this model. However, this is quite painful as the user needs to remember multiple logins and usually go through a log-in procedure at every site, which disturbs the smooth flow of web browsing. What we really want is a

seamless experience, where the person decides which profile she wants to adopt and then websites automatically recognize her and offer her customized content. Ideally, the user should not even be aware that she needs to identify herself to a website, but is naturally offered information relevant to her.

What we need therefore is a user-side personal profile that describes a person and her interests, affiliations and acquaintances in an open standard. Semantic Web [2] technologies provide exactly this in the form of a FOAF (Friend of a Friend) [3] file. In this paper, we propose a personalization mechanism that allows a web server to identify a user's FOAF file with every page request. The web server may then use the information in the FOAF file to personalize the page it returns, potentially retrieving additional information on the web in the process. This is where the advantages of using Semantic Web technologies become apparent. Since the FOAF file is in RDF [1], the information it contains can easily be combined with other RDF information on the web.

In the following, we first discuss the concrete details of our approach (section 2) and then related work on personalization mechanisms (section 3). We then demonstrate the feasibility of our approach through a proof-of-concept implementation in section 4, showing how the AIFB website personalizes its welcome page with information from a visitor's FOAF file. Of course, this is still a fairly simple use of the information. More sophisticated usage of the data in the visitor's FOAF file can enable a range of scenarios as discussed in section 5. Finally, we discuss some of the implications of our approach, in particular with respect to privacy in section 6, before summarizing the contributions of the paper in the concluding section 7.

2 Approach

We rely on a user's Friend Of A Friend (FOAF) [3] profile to personalize her browsing experience. The FOAF vocabulary is one of the most popular RDF vocabularies on the web and describes a person, in terms of several attributes of the person, such as homepages, affiliations, photographs and contact details, as well as specifying the acquaintances and friends of the person. Millions of FOAF files already exist, and are particularly popular within the blogging community.

The FOAF vocabulary definitions are written as RDF [1] statements. This allows software to process the FOAF information and follow links to the FOAF files of friends and acquaintances to gather as much information as it needs. Using RDF as a description format has another advantage: we can easily combine FOAF statements with statements from other RDF vocabularies, such as RSS[1] for describing blog feeds, and the Web of Trust (WOT) Schema, to describe signatures on RDF documents, or geographical location vocabularies. Thus, a FOAF file can contain all kinds of information about the user and point to files with more specific information about the user. The web server can use as much of the information as it understands, ignoring the rest. So the system allows

[1] `http://en.wikipedia.org/wiki/RSS_(protocol)`

for a very graceful performance degradation, and still enables it to express any information that could be useful for personalization.

To give web servers access to a user's FOAF profile, we extend HTTP's `GET` method to include a new parameter that points to the URL of the user's FOAF file. When the web server receives a `GET` request, it examines its parameters in order to determine which information to send back. At the most obvious level, this includes the web page to return. However, there is also an element of browser-personalization involved. For example, depending on which browser is requesting the page, Internet Explorer or a Mozilla browser, the web server can present different versions of the web page best suited to the browser. By extending the `GET` parameter to point to a user's FOAF file, the web server can fetch information about the user and customize the returned page for the user.

This is a fairly simple idea, but it is attractive for several reasons:

1. By relying on the HTTP protocol, we are using the lowest common denominator when it comes to web access, so this method can be used with minimal modifications by most websites.
2. This eliminates the need for the user to login to a site explicitly, with all its associated problems of remembering the login/password details, the time and effort required for the login procedure as well as the break in the smooth navigation through the web.
3. Personal information is within the control of the user rather than multiple websites, so the user can decide what information to expose. Unlike previous user passport methods [12], the FOAF files are simple to understand, straightforward to create, and based on an open standard format.
4. The FOAF file is passed with every HTTP `GET` call, enabling the user to change profiles within a given session.

3 Related work

There are several mechanisms used currently for achieving personalization on the web [9]. These include *server-side accounts*, which require the user to create an account on the website and log in to it when making use of personalized services, *cookies*, used for storing identification and user preferences on the user's machine, and *identity profiles*, such as *Microsoft Passport* [12], *AOL Screen Name*[2] or *OASIS Open Identity*, which provide a single sign-on for multiple services.

Server-side accounts are provided by most major websites and portals such as *Amazon* and *Google*, and cookies are used (often indiscriminately) by an even larger number of websites. However, there are distinct disadvantages to all of these approaches. To begin with, none of these mechanisms give fine-grained control to the user of the information he or she is presenting to a website. Server-side accounts usually require a standard list of information regardless of whether and how it is used subsequently. Furthermore, there is no single point of control for the user. With server-side accounts, user information is scattered all over

[2] `https://my.screenname.aol.com/`

the web, tied up in hundreds of websites for the typical web surfer. Cookies, simple site-defined key-value pairs, do store information on the user end, but since the format is defined by the web site, more often than not the cookies are meaningless to the user. Since cookies are site-specific user information, often the same information, such as the geographical location of the user, is stored in multiple cookies for different websites. Identity profiles do present an unified store of user information, but these are typically stored in proprietary formats at the vendor end, such as *Microsoft* or *AOL*, thus again compromising user control of her information. With our approach based on FOAF files, information is stored in open standards on the user end, allowing for user control of a centralized store of her information. In addition, since FOAF uses Semantic Web standards, FOAF information can easily be combined with other Semantic Web information, further enhancing its value.

Secondly, the majority of these personalization mechanisms do not store social information about friends, acquaintances and colleagues. Assuming the user has made extensive use of social community sites, such information is locked in multiple websites, such as *LinkedIn*[3], *OpenBC*[4] and *Yahoo!*[5]. This means that the user has to tediously enter the same people in multiple websites and the communities often have different kinds of contacts (for example, *OpenBC* is used extensively and and almost exclusively in Germany), which are not connected to each other. The FOAF files allow us to not only present a single-point for describing all contacts, but also enable combining this information with the web browsing behavior, so, for instance, a web site can tell you which of your friends visited the site recently.

Despite the single point of control, our approach does not eliminate the need for site-specific user profiles and preferences. Despite generally preferring large text sizes, a user may still want to specify that she prefers the standard text size on *Yahoo!*. Such information may still need to be stored at the web site end. However, our approach does mean that the user always has a set of default preferences and the site only needs to save information about site-specific preferences that differ from the default preferences. The primary criticism of user-centric identity management has been that it is not portable [11]. Personal information stored on one computer cannot be easily transferred to my mobile phone. This is a non-issue in our case, since the identity as FOAF file is always accessible over the web. Of course, this can lead to security and access control issues, which we refer to in the discussion (section 6).

4 Implementation

Extending the HTTP `GET` request requires modifications on both the client and the server end. At the client end, the web browser must be extended to include a link to the user's FOAF file when sending a `GET` request to a web server. At

[3] `https://www.linkedin.com/`
[4] `https://www.openbc.com`
[5] `http://www.yahoo.com`

the server end, the web server must be able to understand the HTTP protocol extension and read the FOAF file pointed to. In the following, we discuss our proposed HTTP protocol extension, the implementation of the extension for the Mozilla-based Firefox web browser[6] and the modification of the AIFB Semantic Portal[7] [7] to make use of the extension.

We extended the HTTP protocol in conformance with the HTTP specification RFC2616 [6]. We added a new line, a property `XFOAF`, that specifies the URL of the user's FOAF file [3]. A web server that is aware of this extension may then fetch the FOAF file, require additional resources if needed, and then process it appropriately. All other web servers can simply ignore the line, since according to RFC2616, unrecognized header fields should be ignored and must be forwarded transparently. Thus the user experience does not change in any way when visiting pages unaware of this extension. Figure 1 shows an example of an extended `GET` call.

```
http://www.aifb.uni-karlsruhe.de/english

GET /english HTTP/1.1
Host: www.aifb.uni-karlsruhe.de
Accept: text/xml,application/xhtml+xml,text/html
XFOAF: http://nodix.de/foaf.rdf
```

Fig. 1. An example of an extended `GET` request.

On the browser side, we used the HTTP extension with the Mozilla based Firefox browser. We used the Firefox plugin *ModifyHeaders*[8] to add an `XFOAF` header and an appropriate value. We tested the extended `GET` with several websites, and as expected we did not encounter any problems. Although *ModifyHeaders* already provides us with all required functionality (deactivating the extension, changing the `XFOAF` value on the fly, etc.), the user interface is too technical by far for the casual user.

On the server side, we enhanced the AIFB Semantic Portal, the official website of the AIFB research institute, to make use of the extended HTTP requests. The portal is a ZOPE application[9] and offers several semantic features, like a full export of its data in OWL, using several vocabularies like SWRC [14], FOAF [3], and vCard [5, 8]. The portal also provides a SPARQL endpoint to query the data, and an RSS feed to syndicate news.

Now that the AIFB web server is able to access the user's FOAF file, it checks for persons the user says to know. On the main page of the portal the user is provided with direct links to these persons. From there, she can explore

[6] `http://www.mozilla.org/firefox`
[7] `http://www.aifb.uni-karlsruhe.de`
[8] `http://modifyheaders.mozdev.org/index.html`
[9] `http://www.zope.org`

Fig. 2. The personalized AIFB portal. The visitor is greeted with his name (top), and may find shortcuts to all known persons working at the AIFB (lower right quarter).

the person's publication list, current contact data, given courses, projects the person works on, and so on. Each user who offers the portal appropriate data is rewarded with personalized access and will be able to access the information she looks for much faster than before. Screenshot 2 shows the AIFB portal being personalized for one of the authors.

As the extended `GET` request only provides us with the URL of the FOAF file, we first need to extract the URI of the visitor. We expect the visitor being connected with a *foaf:primaryTopic*-relation to the FOAF URL. Now we can search for information about the visitor, like her name or the persons she knows. The latter needs to be compared to the local knowledge base. Either some persons have the same URI in both the visitor's FOAF file and the server's knowledge base, or based on a inverse functional property (like *foaf:mbox_sha1sum*) we are able to infer the equality of two persons in the two ontologies. We neither force the visitor to use the same URIs as we do, nor to use any URIs at all (and often they do not, because it was considered impolite for some time to assign URIs to persons, and thus inverse functional properties were used for identification). The described extension was implemented in Python[10], using rdflib[11].

[10] `http://www.python.org`
[11] `http://rdflib.net`

5 Use cases

Access to a semantic profile of the person browsing could enable the personalization of the browsing experience in several interesting ways. A number of potential usage scenarios are listed below:

- At a basic level, the web server can use the profile information of the browsing person to prefill forms on the website. Many web forms ask for information such as name, affiliation, homepage, all of which have dedicated tags in the FOAF file. Instead of the user having to manually enter all this information every time for multiple websites, the visited websites could automatically retrieve this information from the user's FOAF file and prefill forms of the website. The user can then review the prefilled fields, modify them if required or add information for 'new' fields.
- A FOAF file also contains links to known persons. By logging visitors and comparing them to the user's acquaintances, she could be notified if someone she knows visited the site recently. As discussed in [16], knowing who I know has visited this site contributes to companionability (doing things with friends), sociability (doing things with people who are similar to me), for establishing authority, and possibly authenticity. Thus, knowing that friends have visited this site increases the value of the site to me, and generally leads to increased site traffic.
- Since my acquaintances are likely to have interests similar to mine, knowledge of which pages they have viewed on a web site can be used to suggest relevant pages to me. At the very least, I know about the goals and interests of my acquaintances and can use this information to better understand the content on the website. This could be especially useful for large web sites with only few pages of real value to me, or if I lack expertise in a particular area and need guidance.
- If the FOAF profile of the browsing user is extended with viewer preference parameters, such as color schemes, preferred languages or accessibility preferences, such as large font size for visibility-impaired users, the web server can automatically customize the served HTML pages for the needs of the user. Such information need not be contained within the FOAF file itself, but it can point to other standardized specifications of user preferences, such as CC/PP [13], and the web server can gather the information from there.

The above usage scenarios can be realized within the current web itself. Given a Semantic Web with richer kinds of information available, additional scenarios become feasible. Thus, knowledge of the tasks and activities of the user as well as knowledge of organizational hierarchies that the user is part of could help the web server to tailor the content it presents to the user.

With more heavyweight infrastructure on the server-side, the web site could support collaborative filtering over communities and topics. One interesting application of our personalization mechanism is the possibility of location-based messages or reminders for friends. For example, given an intranet or project website or just any website that is visited often by a group of friends, I might

want to leave a message to notify me when someone I know has viewed a page or to leave a personalized message for them to be delivered when they access a particular page. This is essentially the virtual equivalent of leaving a post-it note on the community fridge.

Many of these scenarios have been explored in various prototype systems, particularly in the fields of human-computer interaction and computer-supported cooperative work. However, these systems have always been standalone and require non-trivial effort in set-up, meaning that they remain primarily interesting research prototypes. The main contribution of our work is that we propose a fairly simple personalization mechanism that relies on open standards and the Semantic Web, is straightforward to implement and therefore has the potential to be used on a large-scale in the real world outside laboratory settings.

6 Discussion

Users are basically giving up part of their privacy for the features suggested in this paper. However, they are now able to control what information they want to expose in a very fine granularity. The user is able to define several FOAF files easily and to switch between the one used by the browser on the fly, or to deactivate them completely. This way they could use another persona in their leisure time than they do at work. They could also decide to define rules that determine which persona to choose, if any, according to the website to be retrieved.

Besides having different personas, a user may also decide to grant different information about themselves to different websites. A discussion forum about the Lord of the Rings may get a list of all the fantasy articles the user has published, but not her academic merits, whereas a job search portal would be granted access to the latter knowledge but not to the first. To realize this, we cannot rely on simple static FOAF files. Instead we would need to enable each user to set up a web service that would decide, according to the user's settings and the websites description, which information to deliver. For example, if the websites specify their privacy practices using standards such as P3P [4], the user can use this information to decide what information to expose to the website. This goes beyond the proposal given in this paper, but we can easily extend it by simply adding the URI of the web service to the FOAF file. No further extension of the HTTP protocol would be required.

As FOAF files are public files on the web, every user could easily "hijack" the FOAF file of another person. In order to counter that problem, the web server may decide to encrypt each delivered page with the PGP public key of the user which would be included or referenced in the FOAF file. Only a browser that has access to the private key of the user would be able to decrypt the site. This also could replace sign-in systems, and free the user from remembering numerous different passwords, or using the same weak password and login combination in a plethora of different places. In fact, since users often tend to use the same or similar login/password combinations for multiple websites, relying solely on this

authentication method is dangerous. A website provider could potentially use the login/password combination of the website's users to access a large number of other password-protected websites. For example, a merger between services like *flickr* and *Yahoo!* could actually provide the new service provider with a list of login/password combinations that were used in both websites, and thus are potentially used in many other websites as well, like *MSN*. Such knowledge could be harmful for both the user as well as the vulnerable web sites. A trusted infrastructure should therefore be in the interest of both the service providers and the users. By using a public key infrastructure as described above, the overall security of the web for the users would increase. This feature could be used independently from the personalization possibilities described in this paper.

In accordance to the spirit of the web, FOAF files are decentralized entities of varying quality. Some users may decide to offer their name, their interests, knowledge about their social networks and even such things as e-mail or real world addresses. Other users may opt not to to provide any of these, or simply lie about them. But this problem is already manifest on the web. If websites like news providers or public discussion forums ask for such information, they have no possibility to check the correctness of this information besides simple validity checks. Actually, the FOAF approach provides the user with the possibility to lie more consistently (and relieves them from entering the information again and again in different places, possibly forgetting or mixing up details), and thus to build much more trustable personas. This is usually enough for most websites – they often do not need to validate the correctness of the information, but rather they just require a basic means to identify their users and build models of them. That is the reason why cookies are in such wide usage, but cookies have several further flaws: they may be tampered with or 'stolen' by an unauthorized third-party [10]. Also cookies are not easily transferable from one computer to the other, whereas a browser on another computer would only require the URL of the FOAF file in order to personalize all the pages. Both parties, website providers and their users gain from this situation.

It should be noted that simply by using their publicly-accessible FOAF profiles, users do not expose any information to web sites that is not already available to the websites today. A website like *CNN*[12] already asks for the user's name and email for certain services. Equipped with this information they could use a search engine to locate the user's FOAF file and collect the information within. The approach described in this paper makes this connection more explicit, reliable, and gives more control to the user.

7 Conclusions

In this paper, we described a simple, backward-compatible extension to the HTTP protocol that enables a web server to use a visitor's FOAF information for considerable personalization of her web browsing experience. We demonstrated

[12] http://www.cnn.com

the feasibility of the proposed extension by implementing it on a proof-of-concept level for the Firefox browser and the AIFB portal. The information in the FOAF file becomes really exciting when websites can tell you whether your friends have visited the same site or the pages they viewed. We would like to explore this within the context of our own research institute portal and examine the usage of our personalization mechanism in practice. A user study (or extensive adoption of the proposed approach) will be required to understand how much information users are actually willing to provide in such a setting, and how useful this information really is in enhancing their web experience.

Acknowledgements

This work was supported by the European Commission in the IST programme under the SEKT project, Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and by the German Federal Ministry of Education and Research (BMBF) under the SmartWeb project (grant 01 IMD01 A).

References

1. D. Beckett. RDF/XML syntax specification (revised), February 2004.
2. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, (5), 2001.
3. D. Brickley and L. Miller. The friend of a friend (FOAF) vocabulary specification, June 2005.
4. L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The platform for privacy preferences 1.0 (p3p1.0) specification, 2002.
5. F. Dawson and T. Howes. vcard MIME directory profile: Internet RFC 2426, 1998.
6. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1, 1999.
7. J. Hartmann and Y. Sure. An infrastructure for scalable, reliable semantic portals. *IEEE Intelligent Systems*, 19(3):58–65, May 2004.
8. R. Iannella. Representing vcard objects in RDF/XML, 2001.
9. M. Koch and K. Mslein. Identities management for e-commerce and collaboration applications. *Int. Journal of Electronic Commerce (IJEC)*, 9(3):11–29, 2005.
10. D. M. Kristol. Http cookies: Standards, privacy, and politics. *ACM Transactions on Internet Technologies*, 1(2):151–198, 2001.
11. D. Mulligan and A. Schwartz. Your place or mine? privacy concerns and solutions for server and client-side storage of personal information. In *Proceedings of Computers, Freedom and Privacy (CFP)*, Toronto, ON, Canada, April 2000.
12. `https://accountservices.passport.net`.
13. F. Reynolds, J. Hjelm, S. Dawkins, and S. Singhal. Composite capability/preference profiles (CC/PP): A user-side framework for content negotiation, July 1999.
14. Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. The SWRC ontology. In *Proc. of the 12th Portuguese Conf. on AI (EPIA 2005)*, volume 3803 of *LNCS*, pages 218–231, Covilha, Portugal, 2005. Springer.
15. M. Svensson. *Defining, Designing and Evaluating Social Navigation*. PhD thesis, Stockholm University, Department of Computer and Systems Sciences, 2002.
16. A. Wexelblat and P. Maes. Footprints: History-rich tools for information foraging. In *Proceedings of CHI 1999*, pages 270–277, 1999.