

A Thin-Server Approach to Ephemeral Web Personalization Exploiting RDF Data Embedded in Web Pages

Dario De Nart, Carlo Tasso, Dante Degl'Innocenti

Artificial Intelligence Lab
Department of Mathematics and Computer Science
University of Udine, Italy
{dario.denart,carlo.tasso}@uniud.it, dante.deglinnocenti@spes.uniud.it

Abstract. Over the last years adaptive Web personalization has become a widespread service and all the major players of the WWW are providing it in various forms. Ephemeral personalization, in particular, deals with short time interests which are often tacitly entailed from user browsing behaviour or contextual information. Such personalization can be found almost anywhere in the Web in several forms, ranging from targeting advertising to automatic language localisation of content. In order to present personalized content a user model is typically built and maintained at server-side by collecting, explicitly or implicitly, user data. In the case of ephemeral personalization this means storing at server-side a huge amount of user behaviour data, which raises severe privacy concerns. The evolution of the semantic Web and the growing availability of semantic metadata embedded in Web pages allow a role reversal in the traditional personalization scenario. In this paper we present a novel approach towards ephemeral Web personalization consisting in a client-side semantic user model built by aggregating RDF data encountered by the user in his/her browsing activity and enriching them with triples extracted from DBpedia. Such user model is then queried by a server application via SPARQL to identify a user stereotype and finally address personalized content.

Key Words: User Modeling, Open Graph protocol, Ephemeral Personalization, RDFa, DBpedia, Privacy

1 Introduction

Personalization is one of the leading trends in Web technology today and is rapidly becoming ubiquitous on the Web. Most of the times the process is evident, for instance when web sites require their users to sign in and ask for their preferences in order to maintain accessible and persistent user profiles. But in other cases personalization is more subtle and it is hidden to the user.

Ephemeral personalization [13], for instance, aims at providing personalized content fitting only short-term interests that can expire after the current navigation session. Most of the times such kind of personalization does not require the user to sign in, since all the information needed to determine which content should be presented may be found in his/her browsing cache and/or content providers are not interested in modelling and archive such short-term interests. An example of ephemeral personalization is targeted advertising, that is providing personalized ads to users as they browse the Web. This task is currently accomplished by checking which cookies are present in the client's browser cache and selecting candidate ads accordingly. This process, however, in most cases results in a particular ad from a previously visited site, "stalking" in such way the user throughout all his/her browsing activities. As the authors of [10] suggest, this may generate a revenue for the advertiser by encouraging customers to return, but can also be extremely annoying and the users may perceive their privacy attacked. Other forms of ephemeral personalization are guided by contextual information derived from the IP address of the client or by analysing the content of the pages that the client requests, like in Amazon's product pages: these are very shallow forms of personalization and do not involve a persistent, long term, user model.

In this work, we propose another way to address ephemeral Web personalization. Our approach consists in collecting semantic metadata contained in visited web pages in order to build a client-side user model. Such model is then queried by content providers to identify a user stereotype and consequently recommend content. By doing this the user has total control over his/her user model and the content provider does not need to save and maintain user profiles, therefore privacy risks are significantly reduced.

Before proceeding forth into the technical matter we would like to point out that our approach heavily relies on the availability of semantic metadata embedded in Web pages: the more metadata available, the more detailed the user profile, vice versa, if visited pages do not contain metadata, no user profile can be built. Luckily, a huge number of Web sites actually provides semantic annotations, consisting of Microformats, Microdata, or RDFa data, mostly conformed to vocabularies such as Facebook's Open Graph protocol, hCard, and Schema.org.

The rest of the paper is organized as follows: in Section 2 we briefly introduce some related works; in Section 3 we present the proposed system; in Section 4 we illustrate our data model; in Section 5 we discuss some experimental results and, finally, in Section 6 we conclude the paper.

2 Related Work

Several authors have already addressed the problem of generating, parsing, and interpreting structured metadata embedded in Web sites. Several metadata formats aimed at enriching Web pages with semantic data have been proposed and adopted by a wide range of Web authors. Microformats¹ such as *hCard*, proposed in 2004, extend HTML markup with structured data aimed at describing the content of the document they are included in. Although similar under several aspects, Microformats are not compliant with the RDF data model, which can be fully exploited by using RDFa data. Facebook's *Open Graph Protocol*², is an RDFa vocabulary proposed in 2010 and has become extremely widespread on the Web due to the fact that it makes possible establishing connections between Web contents and any Facebook user's social graph. However, it also caused many concerns about its privacy and service-dependency issues [19]. Another relevant format is *schema.org*³, proposed in 2011 and supported by several major search engines such as Google, Yahoo, Yandex, and Bing. Schema.org data can be expressed as RDFa or *Microdata*, an HTML specification aimed at expressing structured metadata in a simpler way than the one provided by Microformats and RDFa.

The authors of [2] provide an extensive survey of the presence of such metadata on the Web based on the analysis of over 3 billion Web pages, showing how, despite being Microformats still the most used format, RDFa is gaining popularity, severely outnumbering Microdata. Moreover, they measured that 6 out of the 10 most used RDFa classes on the Web belong to the Open Graph protocol.

Automatic metadata generation has been widely explored and can be achieved in many ways: extracting entities from text [6], inferring hierarchies from folksonomies [18], or exploiting external structured data [11]. Interoperability issues among various metadata formats have been discussed as well: for instance, the authors of [1] propose a metadata conversion tool from Microformats to RDF.

Other authors have discussed how Semantic Web tools, such as ontologies and RDF, can be used to model users' behaviours and preferences in Recommender Systems [7]. However, the field on which most research efforts are focused is Personalized Information Retrieval. For instance in [17] is presented an approach towards Ontological Profile building exploiting a domain ontology: as the user interacts with the search engine, interest scores are assigned to concepts included in the ontology with a spreading activation algorithm. The authors of [5] discuss a system that builds a user model aggregating user queries raised within a session and matching them with a domain ontology. Finally, the authors of [4]

¹ <http://microformats.org/>

² <http://ogp.me/>

³ <http://schema.org/>

and [14] suggest that ontological user models can be built as “personal ontology views”, that are projections of a reference domain ontology derived by observing user interest propagation along an ontology network. However, in all these works, user profiles are specializations or projections of a domain ontology and therefore their effectiveness relies on the availability, scope, and quality of such pre-existing asset.

The problem of preserving users’ privacy while providing personalized content, presented in [16] and recently extensively surveyed in [9], has been widely discussed in literature and many authors tried to address it. The authors of [12] show how to adapt the leading algorithms in the Netflix Prize in order to achieve differential privacy, that is the impossibility of deducing user data by observing recommendations. The authors of [3] propose a different approach in which part of the user data remains at client side in order to preserve user privacy. Personal Data Store applications, such as OpenPDS ⁴, instead provide a trusted, user controlled, repository for user data and an application layer which can be used by service providers to address personalized content without violating users’ privacy. Finally, a recent patent application [20] also claims that the so-called targeting advertising can greatly benefit from the use of semantic user models extracted from Web usage data. The authors, however, do not provide any hint on their extraction technique, focusing, instead, on the architecture and deployment issues of their system.

3 System Architecture

In order to support our claims, we developed an experimental system consisting in a client and a server module built using well-known open source tools such as Apache Jena and Semargl. Figure 1 shows the workflow of the system. The basic idea behind our work is that user interests can be identified by observing browsing activity and by analysing the content of visited Web sites, thus our goal is to exploit the user himself as an intelligent Web crawler to provide meaningful data for building his/her personal profile, therefore the project was named *Users As Crawlers* (herein *UAC*). A compact OWL2 ontology, herein referred as *UAC ontology*, was developed as well in order to introduce new modelling primitives and to support classification of Web pages. Among others, the primitives defined in the UAC ontology are: *relatedTo*, which associates Web pages with DBpedia entities named in the metadata, *nextInStream*, which associates a page with the next one visited by the user, and *previousInStream*, which is the inverse of *nextInStream*.

The client module handles user modelling: it includes three modules, a *Metadata Parser*, a *Data Linker*, a *Reasoner*, and a compact triplestore. The Metadata Parser reads the header sections of the visited web pages and extracts RDF triples from available RDFa metadata. Due

⁴ <http://openpds.media.mit.edu>

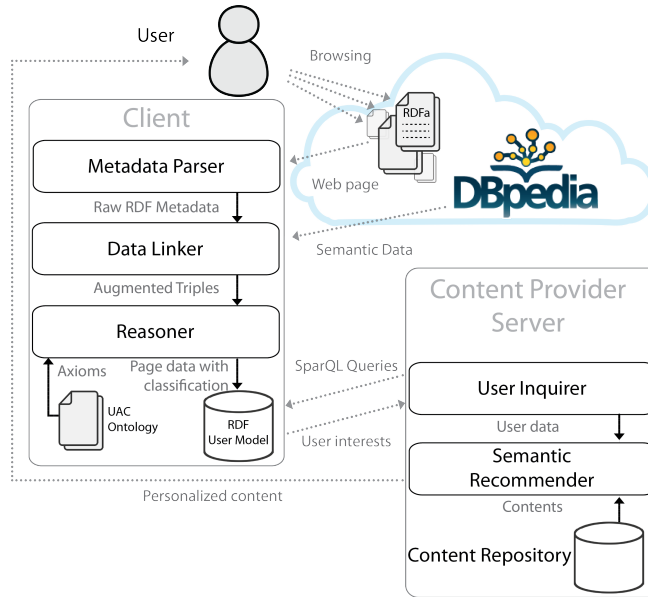


Figure 1. Work flow of the System.

to its large availability, the preferred metadata format is OpenGraph RDFa, however other formats are allowed as well, as long as they can be converted into RDF. The Data Linker receives the collected triples as input and adds new triples by linking visited pages with DBpedia entities. This task is accomplished by both expanding URIs pointed by object properties and by analysing the content of datatype properties such as *tag*, *title*, and *description* in order to find possible matches with DBpedia entries. A list of stopwords and a POS Tagger are used by the Data Linker to identify meaningful sequences of words to be matched against DBpedia. Finally, the augmented set of triples is processed by a Reasoner module, performing logic entailments in order to classify visited pages according to the Open Graph protocol, the DBpedia ontology, and the UAC ontology. In our prototype the reasoning task is performed by the OWL Lite Reasoner that comes bundled with Apache Jena, but any other OWL Lite or DL reasoner (e.g: Pellet) could fit as well. The result of this process is a RDF user model, built incrementally as the user visits Web pages, in which visited pages are classified by *rdf:type* properties and have a hopefully high number of semantic properties linking them each other and to DBpedia. In our prototype system the client is a standalone application, however, in a production scenario it could be a Web browser plug-in, in order to incrementally build the user profiles as pages are downloaded by the Web browser. Since the client contains the user model, it also allows control over it: the user can decide which pages

to be included in the model and whether to maintain crawled data or not between different browsing sessions. Moreover, the user model can be exported at any time to a RDF-XML file to be inspected; though currently the client has no user model visualization module, several visualization and editing tools are available, such as Protege⁵.

The server part of the system is designed to simulate a content provider scenario and consists in two modules, a *Semantic Recommender*, a *User Inquirer*, and a content repository. The goal of the server is to identify a user stereotype in order to suggest to the user a relevant content, but instead of maintaining user profiles in a repository, it just “asks” to connected clients if their user models have some characteristics, much like the *Guess Who* game. Since all the user modelling duties are left to the client, the server module is particularly lightweight and, therefore, we defined it a *thin server* to emphasize the role reversal with respect to the traditional Web personalization approach. We assume each content to be addressed towards a specific user stereotype, which is a realistic assumption since many e-commerce companies already perform market segmentation analysis. We exploit such knowledge in order to map user characteristics into a specific stereotype and therefore contents to be recommended. More specifically, in our current experimental system we use a decision tree for classifying the user, as shown in Figure 2. Each node is associated with a specific SPARQL query and each arc corresponds to a possible answer to the parent node’s query. Stereotypes are identified on the leaves of the tree. When a client connects, it receives the SPARQL

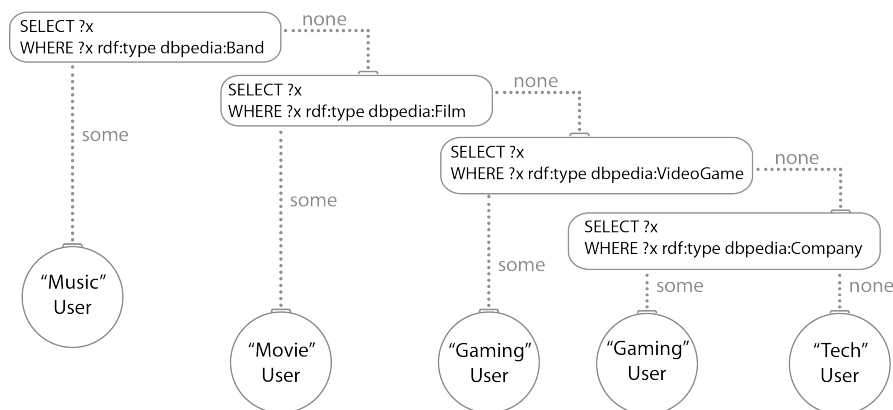


Figure 2. A decision tree with SPARQL queries on the nodes and user stereotypes on the leaves.

query associated with the root node in order to check whether a specific

⁵ <http://protege.stanford.edu/>

characteristic is present in the user model. The Semantic Recommender module handles the client’s answer to the query and when it receives a positive answer fetches content or, if the answer is negative, further queries are proposed until a user stereotype can be identified. Due to the hierarchical nature of the decision tree, we expect the number of queries to be asked to the client to be very small: indeed, in our experimental setting in the worst case six queries were needed. In order to preserve users’ privacy, the server does not have full access to user RDF data: though any SPARQL query is allowed, the client returns only the number of Web pages matching the query, therefore their URIs are unknown to the server.

4 Data linking and classification

In order to better present our user modelling technique, in this section we are showing a step-by-step example of what the proposed system does when a Web page is visited. In this example we are considering a randomly chosen Rottentomatoes.com page and in Figures 3, 5, and 6 extracted and inferred data are shown in the RDF-XML syntax. As the page is loaded, the Metadata Parser retrieves the available RDFa metadata. Metadata commonly embedded in Web pages actually provide a very shallow description of the page’s content: the Open Graph protocol itself specifies only four properties as mandatory (*title*, *image*, *type*, and *url*) and six object classes (video, music, article, book, profile, and website). However, these information constitutes a good starting point, especially when a few additional, but optional, properties (e.g.: *description*, *video:actor*, and *article:tag*) are also specified, which can provide URIs of other entities or possible “hooks” to more descriptive ontologies. In Figure 3 is shown the metadata available in the Web page, including both mandatory and optional Open Graph properties. The next mod-

<pre> <rdf:Description rdf:about="http://www.rottentomatoes.com/m/mad_max/"> <og:url rdf:resource="http://www.rottentomatoes.com/m/mad_max/"> <og:type xml:lang="en">video.movie</og:type> <og:image rdf:resource="http://content6.flixster.com/movie/11/17/23/11172324_800.jpg"/> <og:title xml:lang="en">Mad Max</og:title> </pre>	<div style="border-left: 1px solid black; border-right: 1px solid black; height: 40px; margin: 0 auto; width: 20px;"></div>	<p>Mandatory Open Graph properties</p>
<pre> <og:description xml:lang="en">Staging the improbable car stunts... </og:description> <video:release_date xml:lang="en">1979-04-12</video:release_date> <video:duration xml:lang="en">93</j.2:duration> <video:director rdf:resource="http://www.rottentomatoes.com/celebrity/1042523-george_miller/"> <video:actor rdf:resource="http://www.rottentomatoes.com/celebrity/joanne_samuel/"> <video:actor rdf:resource="http://www.rottentomatoes.com/celebrity/mel_gibson/"> <video:actor rdf:resource="http://www.rottentomatoes.com/celebrity/1008104-hugh_keaysbyrne/"> <video:actor rdf:resource="http://www.rottentomatoes.com/celebrity/steve_bisley/"> <video:actor rdf:resource="http://www.rottentomatoes.com/celebrity/roger_ward/"> </rdf:Description> </pre>	<div style="border-left: 1px solid black; border-right: 1px solid black; height: 40px; margin: 0 auto; width: 20px;"></div>	<p>Optional Open Graph properties</p>

Figure 3. Metadata extracted from the example Web page

elling step, performed by the Data Linker module, aims at enriching such data by linking the visited page to possibly related ontology entities. This step needs a reference ontology and we adopted a general purpose and freely available reference ontology, i.e. DBpedia. This choice is motivated by three factors: (i) in a realistic scenario it is impossible to restrict users' Web usage to a particular domain, (ii) authors may describe their contents in ways non compliant to a single taxonomy crafted by a domain expert, therefore, the ontology needs to be the result of a collaborative effort, and (iii) since the modelling task is to be accomplished at client-side, we need a freely accessible ontology.

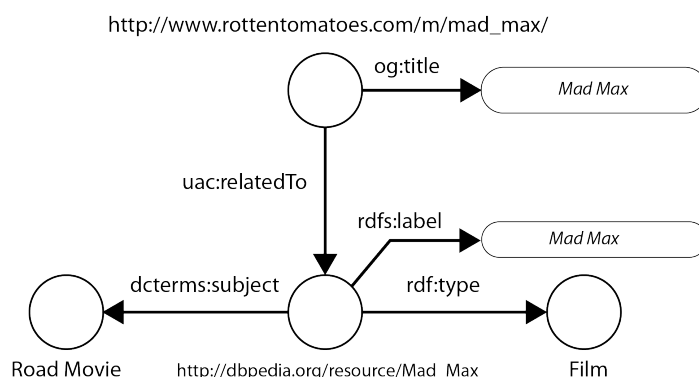


Figure 4. An example of DBpedia data linking

The Data Linker analyses the RDF data extracted from the pages in order to discover “hooks” to DBpedia, that are named entities present in DBpedia mentioned in the body of properties such as *title* or *description*. Such properties are analysed by means of stopwords removal and POS tagging to find possible candidate entities; candidate entities are then matched against DBpedia entries to get the actual ontology entities. As shown in Figure 4, in our example the association between the value of the *title* property and the *Mad_Max* DBpedia entity is trivial, since the value of the *title* property and the one of the *rdfs:label* property of the DBpedia entity are the same string. However there may be more complex cases: for instance the title “Annual Open Source Development Survey” can provide a hook for the *Open_Source* entity.

Once these entities have been identified, they are linked to the Web page RDF representation with a *relatedTo* property, defined in the UAC Ontology. An additional UAC property, *prevInStream*, containing a link to the page that precedes the considered one in the navigation stream, is added to the page data as shown in Figure 5. All the *rdf:type*, *dc:subject*, and *db:type* attributes of the linked DBpedia entity are then imported into the RDF user model, in order to provide further information about


```

<rdf:Description rdf:about="http://www.rottentomatoes.com/m/mad_max/">

  Crawled metadata

  <uac:relatedTo rdf:resource="http://dbpedia.org/resource/Mad_Max"/>
  <uac:relatedTo rdf:resource="http://dbpedia.org/resource/director"/>
  <uac:relatedTo rdf:resource="http://dbpedia.org/resource/Miller"/>
  <uac:relatedTo rdf:resource="http://dbpedia.org/resource/car"/>
  <uac:relatedTo rdf:resource="http://dbpedia.org/resource/stunts"/>
  <uac:relatedTo rdf:resource="http://dbpedia.org/resource/George_Miller"/>

  <uac:prevInStream rdf:resource="http://www.google.it/url?sa..."/>
</rdf:Description>

```

Related DBpedia entries

Previously visited page

Figure 5. The example page linked to related DBpedia entities and the previously visited page

the contents of the page and to support the classification task.

The final step of the modelling activity is the classification performed asynchronously by the *Reasoner*, which integrates the data gathered in the previous steps with class and property axioms provided by the Open Graph protocol and by the UAC ontology. Due to its asynchronous nature, the Reasoner can also link a page to the one subsequently visited using the *nextInStream* property. The result, as shown in Figure 6, is, aside from the *nextInStream* property, a series of *rdf:type* properties which provide a classification of the visited page. For instance, in our example the entailed properties state that the URI *http://www.rottentomatoes.com/m/mad_max/* corresponds to a *Website*, a *Work* (as defined in DBpedia), and a *Film*. Such type properties are added to the RDF representation of the Web page, along with the crawled data and the DBpedia data generated in the previous steps, and stored in the user model.

```

<rdf:Description rdf:about="http://www.rottentomatoes.com/m/mad_max/">

  Crawled metadata
  Related DBpedia entries
  Previously visited page

  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdf:type rdf:resource="&uac;Website"/>
  <rdf:type rdf:resource="http://dbpedia.org/ontology/Work"/>
  <rdf:type rdf:resource="http://dbpedia.org/ontology/Film"/>

  <uac:nextInStream rdf:resource="http://www.rottentomatoes.com/m..."/>
</rdf:Description>

```

Inferred rdf:type properties

Subsequent visited page

Figure 6. The visited page data annotated with the properties inferred by the Reasoner.

The user model is built incrementally as Web pages are visited and can be queried at any time using the SPARQL query language. This choice allows the server to formulate an extremely wide range of queries, from very generic to utterly specific, and to achieve an arbitrary level of detail in user stereotyping. For instance, the content provider may be interested in recommending just movies rather than books and, by asking for the Web pages with a *rdf:type* property set to “Film”, it will detect, in our example, that the user visited a page about a movie. In another scenario, the content provider may be interested in determining exactly which kind of movie to recommend and by asking for all the Web pages *related to* a DBpedia entity with a *dcterms:subject* property set to “Road Movie” it will detect that the user visited a page about a road movie. Countless other examples are possible: for instance a content provider might be interested in knowing if the user has visited sequentially a given number of sites dealing with the same topic or if he/she has ever visited a page dealing with multiple given topics.

5 Evaluation

Formative tests were performed in order to evaluate the accuracy of the proposed method. In our experiment, we asked a number of volunteers (mostly university students) to let us use their browsing histories, in order to have real-world data. To avoid biases, browsing data was extracted only from sessions occurred in the five days before the test was performed. All test subjects were completely unaware of the real purpose of the experiment. After supplying the data, volunteers were asked to review their own browsing history in order to identify different sessions and to point out what they were actually looking for. At the end of the interviews we were able to identify six user stereotypes, much like market analysts do when performing segmentation analysis. Since we had no real content to provide in this experiment, we only classified users. The six identified stereotypes are: (i) people mostly interested in economics (nicknamed *business*), (ii) mostly interested in courses, seminars, summer schools, and other educational events (*student*), (iii) mostly interested in films and tv series (*moviegoer*), (iv) mostly interested in music (*musician*), (v) mostly interested in videogames (*gamer*), and, finally, (vi) people whose main interests are hardware, programming, and technology in general (*techie*). Four iterations of the data gathering and testing process were performed, each time with different volunteers, in order to incrementally build a data set and to evaluate our approach with different users with different browsing habits and different size of the training set. In the first iteration 36 browsing sessions were collected and labelled, in the second 49, in the third 69, and in the fourth 82.

All RDFa data included in the Web pages visited by our test users was considered and used by our test prototype to build a RDF user

model for each browsing session. Over the three iterations, the average number of Web sites visited in a single browsing session was 31.5 and the average number of RDF triples extracted from a browsing session after the Data Linker performed its task was 472.8, that is an average 15 triples per page, which actually provides a significant amount of data to work on.

During each iteration of the evaluation, the *rdf:type* properties of the visited Web pages were considered as features and used to train a Decision Tree. In this experiment the J48 algorithm [15] was used; in Figure 7 we show an example of a generated tree, built during the third iteration. The nodes of the tree were then replaced with SPARQL queries and then this structure was used to classify a number of user models. Due to the shortage of test data, a ten-fold cross validation approach was used to esteem the accuracy of the system. Table 1 shows the re-

```

Band <= 0
| OfficeHolder <= 0
| | VideoGame <= 0
| | | Person <= 0
| | | | Magazine <= 0
| | | | | EducationalInstitution <= 0: techie (29.0/15.0)
| | | | | EducationalInstitution > 0: student (2.0)
| | | | Magazine > 0: gamer (3.0/1.0)
| | | Person > 0
| | | | Organisation <= 0: moviegoer (12.0/2.0)
| | | | Organisation > 0
| | | | | Magazine <= 0: moviegoer (2.0/1.0)
| | | | | Magazine > 0: business (2.0)
| | VideoGame > 0: gamer (5.0)
| OfficeHolder > 0: business (5.0)
Band > 0: musician (9.0)

```

Figure 7. A decision tree built during the third iteration of the experiment

sults of the classification over the four iterations of the data set. Our system was compared with the ZeroR predictor, which always returns the most frequent class in the training set in order to have a baseline. For this formative experiment, only the precision metric (defined as the number of correctly classified sessions over the total number) was considered. Though precision values are not very high, it is important to point out two limitations of the performed tests: the number of considered browsing sessions is extremely low, due to the fact that only a handful of volunteers let us analyse and use freely their browsing history data; in fact many volunteers dropped out as soon as they realized that their actual browsing history and not some laboratory activity was needed. Secondly, these results were obtained by considering only the *rdf:type* attribute as feature when building the decision tree. Evaluation and de-

Table 1. Average precision of the UAC system and of a ZeroR classifier on the considered data sets.

Data Set size	ZeroR precision	Tree precision
36	0,306	0.639
49	0,195	0.601
69	0,217	0.623
82	0,203	0.631

velopment are ongoing and further experiments, with more test users, more stereotypes, and a richer RDF vocabulary are planned.

6 Conclusion and Future Work

In this paper we presented a new approach towards ephemeral personalization on the Web, relying on semantic metadata available on the web and, even though the presented results are still preliminary, the overall outcome is very promising. With the growth of the Web of Data, we expect in the next few years to be able to raise the average number of extracted triples from a browsing session and therefore build more detailed user profiles.

In our opinion this approach could fit particularly well to the application domain of targeted advertising because of three major advantages over the actual cookie-based techniques: (i) our approach can recommend novel contents related to current browsing context, rather than associate a user with a set of already visited (and potentially disliked) contents, (ii) the explicit decision model of the decision tree can easily be reviewed by domain experts, supporting market analysis and knowledge engineering, and (iii) by deploying the user model at client side, the user has total control over his/her own data, addressing many privacy concerns. However, the proposed approach has one major drawback: in order to receive personalized contents, users have to install a client, which may be either a browser plug in or a standalone application. Anyway, this seems to be necessary for providing real privacy and also other works aimed at addressing the privacy issues of online advertising have stated the need of a software agent [8]. Our future plans include, among other extensions, the integration of a Keyphrase extraction module aimed at automatically extracting significant phrases from textual data included in Web pages, and enrich in such way the content metadata available for the Reasoner and Recommender modules [6]. Future work will also address scalability issues, possibly replacing some of the currently employed libraries with ad-hoc developed modules.

References

1. Adida, B.: hgrddl: Bridging microformats and rdfa. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(1), 54–60 (2008)
2. Bizer, C., Eckert, K., Meusel, R., Mühleisen, H., Schuhmacher, M., Völker, J.: Deployment of rdfa, microdata, and microformats on the web—a quantitative analysis. In: *The Semantic Web—ISWC 2013*, pp. 17–32. Springer (2013)
3. Castagnos, S., Boyer, A.: From implicit to explicit data: A way to enhance privacy. *Privacy-Enhanced Personalization* p. 14 (2006)
4. Cena, F., Likavec, S., Osborne, F.: Propagating user interests in ontology-based user model. *AI* IA 2011: Artificial Intelligence Around Man and Beyond* pp. 299–311 (2011)
5. Daoud, M., Tamine-Lechani, L., Boughanem, M., Chebaro, B.: A session based personalized search using an ontological user profile. In: *Proceedings of the 2009 ACM symposium on Applied Computing*. pp. 1732–1736. ACM (2009)
6. De Nart, D., Tasso, C.: A domain independent double layered approach to keyphrase generation. In: *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies*. pp. 305–312. SCITEPRESS Science and Technology Publications (2014)
7. Gao, Q., Yan, J., Liu, M.: A semantic approach to recommendation system based on user ontology and spreading activation model. In: *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*. pp. 488–492 (Oct 2008)
8. Guha, S., Cheng, B., Francis, P.: Privad: practical privacy in online advertising. In: *Proceedings of the 8th USENIX conference on Networked systems design and implementation*. pp. 13–13. USENIX Association (2011)
9. Jeckmans, A.J., Beye, M., Erkin, Z., Hartel, P., Lagendijk, R.L., Tang, Q.: Privacy in recommender systems. In: *Social Media Retrieval*, pp. 263–281. Springer (2013)
10. Lambrecht, A., Tucker, C.: When does retargeting work? information specificity in online advertising. *Journal of Marketing Research* 50(5), 561–576 (2013)
11. Liu, X.: Generating metadata for cyberlearning resources through information retrieval and meta-search. *Journal of the American Society for Information Science and Technology* 64(4), 771–786 (2013)
12. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 627–636. ACM (2009)
13. Mizzaro, S., Tasso, C.: Ephemeral and persistent personalization in adaptive information access to scholarly publications on the web. In: *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. pp. 306–316. AH '02, Springer-Verlag, London, UK, UK (2002), <http://dl.acm.org/citation.cfm?id=647458.728228>
14. Osborne, F.: A pov-based user model: From learning preferences to learning personal ontologies. In: *User Modeling, Adaptation, and Personalization*, pp. 376–379. Springer (2013)

15. Quinlan, J.R.: C4. 5: programs for machine learning, vol. 1. Morgan kaufmann (1993)
16. Ramakrishnan, N., Keller, B.J., Mirza, B.J., Grama, A.Y., Karypis, G.: Privacy risks in recommender systems. *IEEE Internet Computing* 5(6), 54–63 (2001)
17. Sieg, A., Mobasher, B., Burke, R.D.: Learning ontology-based user profiles: A semantic approach to personalized web search. *IEEE Intelligent Informatics Bulletin* 8(1), 7–18 (2007)
18. Tang, J., Leung, H.f., Luo, Q., Chen, D., Gong, J.: Towards ontology learning from folksonomies. In: *IJCAI*. vol. 9, pp. 2089–2094 (2009)
19. Wood, M.: How facebook is ruining sharing. Weblog post 18 (2011)
20. Yan, J., Liu, N., Ji, L., Hanks, S.J., Xu, Q., Chen, Z.: Indexing semantic user profiles for targeted advertising (Sep 10 2013), uS Patent 8,533,188