Supporting Global User Profiles Through Trusted Authorities

Ibrahim Cingil
Turkiye Is Bankasi, Head Office
Istanbul, Turkey
Email: ibrahim.cingil@isbank.com.tr

Abstract

Personalization generally refers to making a Web site more responsive to the unique and individual needs of each user. We argue that for personalization to work effectively, detailed and interoperable user profiles should be globally available for authorized sites, and these profiles should dynamically reflect the changes in user interests.

Creating user profiles from user click-stream data seems to be an effective way of generating detailed and dynamic user profiles. However a user profile generated in this way is available only on the computer where the user accesses his browser, and is inaccessable when the same user works on a different computer. On the other hand, the integration of Internet with telecommunication networks have made it possible for the users to connect to Web with a variety of mobile devices as well as desk tops. This requires that user profiles should be available to any desktop or mobile device on the Internet that users choose to work with.

In this paper, we address these problems through the concept of "Trusted Authority". A user agent at the client side that captures the user click stream, dynamically generates a navigational history 'log' file in Extensible Markup Language (XML). This log file is then used to produce the 'user profiles' in Resource Description Framework (RDF). User's right to privacy is provided through the Platform for Privacy Preferences (P3P) standard. User profiles are uploaded to the trusted authority and served next time the user connects to the Web.

The trusted authority concept, serving as a namespace qualifier, provides globally unique userid/password identification for users. Furthermore user profiles dynamically reflect the changes in their interests since the data generated while they are surfing the Web contribute to their profile. Also since the user profiles are defined in RDF, they are interoperable and available to any type of authorized device on the Internet.

1 Introduction

Current approaches to personalization involve a number of techniques. Some require users' active involvement through filling out forms or following a set of questions in the form of a decision tree, which can be a tedious job for the users. Other approaches operate behind the scenes like cookies or server logs, without relying on user-input. Each of these techniques used on the Internet for personalization purposes has some drawbacks. For example, the approaches relying on the user input is static in the sense that the changes in the user interests and preferences are not reflected unless the user updates the form. In the cookie approach, if the user has never visited a site before, no historic data is available. Also cookies from different Web servers have heterogeneities among them and there is no possibility to resolve the schematic conflicts among different cookie types. Hence, a lot of information is being stored with limited use. On the other hand, in the server logs approach, it is hard to distinguish the accesses of a single user to the web site during a session, which is called a user transaction. Current practices used to identify these transactions are either based on assumptions like a single IP is always used by the same user, or through a user login process like 'myYahoo'.

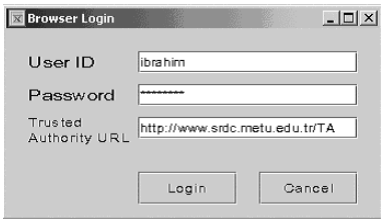

Figure 1: Web browser login window

The personalization architecture presented in [2], which complements the work presented in this paper, provides dynamically created user profiles to the Web servers by making use of the recent data exchange, metadata and privacy standards from the World Wide Web Consortium (W3C). For this purpose, a User Agent at the client side works in close cooperation with the Web browser to obtain the click stream of user actions as he surfs the Web. The user click-stream data (i. e., the user log file) obtained this way is kept as an XML file [9]. The log file contains the subjects of resources the user is interested in, as well as his navigational pattern among the resources. However, since this data is uninterpreted and huge, it is not feasible to use it on the Internet for personalization purposes. Therefore descriptive

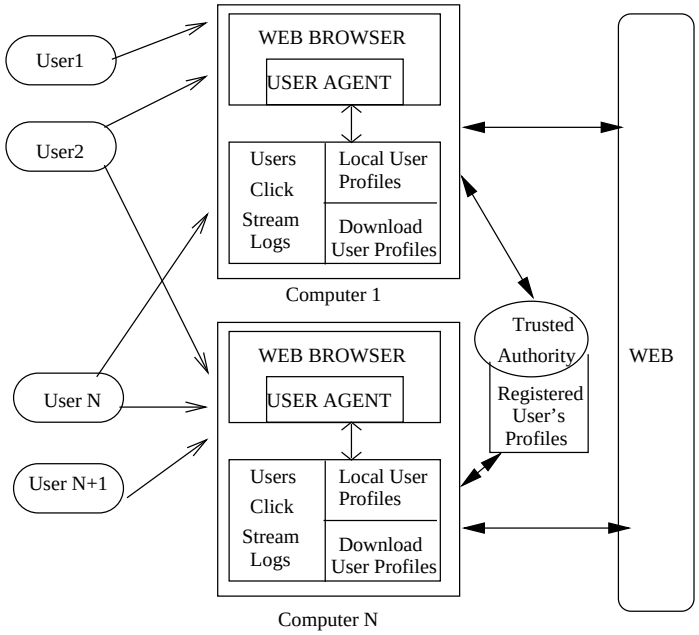

Figure 2: Client side of the architecture

```
<!DOCTYPE LOG [
  <!ELEMENT LOG (LOGENTRY*)>
  <!ELEMENT LOGENTRY (DATE,SERVER,REQUEST?,ACTION,SUBJECT,
  TYPE,DURATION,REFERRER?)
  <!ELEMENT DATE  (DAY,MONTH,YEAR,TIME,TIMEZONE)>
  <!ELEMENT SERVER (NAME,IP,PORT)>
  <!ELEMENT REQUEST (QUERY+)>
  <!ELEMENT ACTION (#PCDATA)>
  <!ELEMENT SUBJECT (#PCDATA)>
  <!ELEMENT TYPE (#PCDATA)>
  <!ELEMENT DURATION (#PCDATA)>
  <!ELEMENT REFERRER (SERVER,SUBJECT,TYPE)>
  <!ELEMENT DAY (#PCDATA)>
  <!ELEMENT MONTH (#PCDATA)>
  <!ELEMENT YEAR (#PCDATA)>
  <!ELEMENT TIME (#PCDATA)>
  <!ELEMENT TIMEZONE (#PCDATA)>
  <!ELEMENT IP (#PCDATA)>
  <!ELEMENT PORT (#PCDATA)>
 <!ELEMENT QUERY (#PCDATA)>
]>
```

Figure 3: DTD of the navigational history log file

statistics are applied to the user log file by executing a standard XML-QL query [4] on it. Since any Web server may use the user profile, the profile information must have a standard format and well-defined meaning. For this purpose, user profiles are generated in RDF [6], [7].

This profile information is valuable, but it cannot be used by companies or service organizations unless a standard mechanism is defined, by which the information can be programmatically accessed on either the client or the server side. Such a mechanism must respect a user's privacy constraints, and P3P standard [5] is used for this purpose. In the P3P architecture, the agents running on both the client side and the server side negotiate using the P3P protocol in order to reach an agreement on the privacy terms of the server according to the user's privacy preferences. Once an agreement is reached the server receives the agreed upon information contained in the user profile.

Although this approach allows creating user profiles dynamically from user click streams, the following problems need to be addressed for effective personalization:

1. More than one person may share the same computer in surfing the Web. In the click stream approach, it is not possible to differentiate the identities of the users.
2. If the same user connects from a different computer to surf the Web, his previous profile is no longer available.
3. With mobile devices becoming highly popular for Internet access, the user may be accessing the Internet with various types of mobile devices. Recognizing the type of the device and providing the profile in the format it accepts, is also important.

The solution to the first problem requires the user to be identified through a login process. Although the current Web browsers do not require a login process, it will be very useful if they do so to easily distinguish click streams of different users. Login information can be stored in the log file, and thus can make it possible to create different user profiles accordingly. This will require the browsers to take the responsibility for handling user logins and the corresponding user profiles.

The second problem requires the profile of a user to be transferred each time the user changes his computer. Transferring the user profiles between computers may not always be possible due to access rights or availability of the previous computer, let alone locating the device. Furthermore transferring profiles does not solve the third problem mentioned; different types of mobile devices require data in different formats.

In this paper we describe an architecture based on the "Trusted Authority" concept to handle the problems mentioned above. The users register to a trusted authority to obtain a login userid and password. The trusted authority in turn keeps the users' profiles as well as their privacy preferences. There can be more than one trusted authority on the Web, and a user may choose to work with any of them.

The paper is organized as follows: Section 2 briefly introduces the standards used in this work. Section 3 discusses the functionality and major components of the proposed system architecture. Finally, Section 4 concludes the paper.

2 An Introduction to the Standards Used

In this section we briefly introduce the Web standards, XML, XML-QL, RDF and P3P used in this work.

XML [9] has gained a great momentum and is emerging as the standard for data exchange on the Internet. XML data is self-describing through content oriented tags and this enables a computer to understand the meaning of data and hence enhances the ability of remote applications to interpret and operate on documents fetched over the Internet. One of the strengths of XML is its extensibility. Anyone can invent new tags for particular subject areas and they define what they mean in document type definitions (DTDs) or similar mechanisms like schemas. But if every business uses its own XML definition for describing its data, it is not possible to achieve interoperability. In other words, a tagged document is not very useful without some agreement among interoperating applications so as to what the tags mean, common DTDs provide for this. A DTD specifies the structure of an XML document by specifying the names of its elements, subelements and attributes.

RDF [6], [7] is a foundation for processing metadata for providing interoperability between applications that exchange machine understandable information and currently is a recommendation by the World Wide Web Consortium (W3C). RDF imposes a syntax and structural constraints in describing resources in order to avoid any ambiguity in expressing meta data so that it becomes machine processable.

The basic data model consists of three object types: resources which are the things being described by RDF, properties which are specific aspects, attributes or relations describing a resource and statements that assign a value to a property of a resource. A resource can be any object that is uniquely identifiable by a Uniform Resource Identifier (URI).

Meaning in RDF is expressed through a reference to an application-specific schema, which defines the terms that will be used in RDF statements and gives specific meanings to them. In other words RDF does not stipulate semantics for each resource description community, but rather provides the ability for these communities to define metadata elements as needed.

The RDF data model provides an abstract, conceptual framework; a concrete syntax is also required and XML is used for this purpose. The XML namespace mechanism serves to identify RDF Schemas.

The Platform for Privacy Preferences (P3P) [5] is a World Wide Web Consortium (W3C) initiative to determine an overall architecture for enabling privacy on the Web. P3P is a specification of syntax and semantics for describing information practices and data elements. The specification uses XML and RDF to capture the syntax, structure, and semantics of the information. The goal of P3P is to enable Web-sites to specify their personal data use and disclosure practices; Web-users to specify their expectations concerning personal data disclosure practices; and software agents to undertake negotiation, on behalf of the parties, in order to reach an agreement concerning the exchange of data between them.

```
WHERE <logentry><duration> $secs </>
                <action> $act </>
      </> CONTENT_AS $le IN "user.log.xml"

CONSTRUCT <total.time.spent> $SUM($secs)  </>
          <total.no.of.visits> $COUNT($le) </>
          <total.no.of.purchases>
 $COUNT($act in {"purchase"}) </>
          <interested.in> <rdf:bag>
          { WHERE <subject> $subj </> IN "user.log.xml"
            CONSTRUCT
  Add_Subject($subj, $COUNT($le),
                      $SUM($secs),
                                $COUNT($act in {"purchase"}) )
          } </rdf:bag></interested.in>

FUNCTION Add_Subject (in $subj, in $tvisits,
      in $totsecs, in $totbuys )
 /* Check if this subject has been processed before,
    if so skip it. */
 { WHERE <subj.processed.list> $subj </> IN "temp.list"
   RETURN
 }
 { /* This subject has appeared for the first time,
   append it to the list of processed subjects */
   CONSTRUCT <subj.processed.list> $subj </>
    APPEND_TO "temp.list"
 }
 { /* Get all the log elements containing this subject
      and create a "rdf:li"
      element that will be placed in the "rdf:bag"
      of the "interested.in" */
   WHERE <logentry><subject> $subj </>
                   <duration> $d </>
                   <action>   $a </>
         </> CONTENT_AS $fle  IN "user.log.xml"
   RETURN <rdf:li><rdf:value> $subj </rdf:value>
          <visit.cnt> $COUNT($fle)  </>
          <interest%> $INTEGER($COUNT($fle) / $tvisits)</>
          <time.in.secs> $SUM($d) </>
          <time.spent> $INTEGER($SUM($d) / $totsecs)</>
          <purchase.cnt> $COUNT($a in {"purchase"})</>
          <purchase%>  $INTEGER($COUNT
         ($a in {"purchase"}) / $totbuys ) </>
          </rdf:li>
 }
END
```

Figure 4: An example XML-QL query to produce user profile in RDF

P3P is designed to help users reach agreements with services (Web sites and applications that declare privacy practices and make data requests). As the first step towards reaching an agreement, a service sends a machine-readable proposal in which the organization responsible for the service declares its identity and privacy practices. A proposal applies to a specific realm, identified by a URI or a set of URIs. The privacy proposal enumerates the data elements that the service proposes to collect and explains how each will be used, with whom data may be shared, and whether data will be used in an identifiable manner. Proposals can

```
<?xml version="1.0" encoding="UTF-8" ?>
<!ENTITY % rdf.common.profile
              SYSTEM "http://www.srdc.metu.edu.tr/user/common.profile.rdf">
<rdf:RDF
     xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#"
     xmlns:rds="http://www.w3.org/TR/WD-rdf-schema#">
  %rdf.common.profile;
  <rdf:Description about="userX">
    <rdf:type resource="http://www.w3.org/TR/WD-rdf-schema#Class"/>
    <rds:subClassOf resource="#User"/>
    <rds:label>userX</rds:label>
    <rds:comment>One of the Users who surfs the Web</rds:comment>
    <total.time.spent>     35 </total.time.spent>
    <total.no.of.visits>    7 </total.no.of.visits>
    <total.no.of.purchases> 1 </total.no.of.purchases>

    <interested.in> <rdf:bag>
        <rdf:li> <rdf:value> book </rdf:value>
                 <visit.cnt>     1 </visit.cnt>      <interest%>    14 </interest%>
                 <time.in.secs>  2 </time.in.secs>   <time.spent%>   6 </time.spent%>
                 <purchase.cnt>  0 </purchase.cnt>   <purchase%>     0 </purchase%> </rdf:li>
        <rdf:li> <rdf:value> history </rdf:value>
                 <visit.cnt>     1 </visit.cnt>      <interest%>    14 </interest%>
                 <time.in.secs>  2 </time.in.secs>   <time.spent%>   6 </time.spent%>
                 <purchase.cnt>  0 </purchase.cnt>   <purchase%>     0 </purchase%> </rdf:li>
        <rdf:li> <rdf:value> military </rdf:value>
                 <visit.cnt>     1 </visit.cnt>      <interest%>    14 </interest%>
                 <time.in.secs>  2 </time.in.secs>   <time.spent%>   6 </time.spent%>
                 <purchase.cnt>  0 </purchase.cnt>   <purchase%>     0 </purchase%> </rdf:li>
        <rdf:li> <rdf:value> weapon </rdf:value>
                 <visit.cnt>     2 </visit.cnt>      <interest%>    30 </interest%>
                 <time.in.secs>  4 </time.in.secs>   <time.spent%>  11 </time.spent%>
                 <purchase.cnt>  0 </purchase.cnt>   <purchase%>     0 </purchase%> </rdf:li>
        <rdf:li> <rdf:value> primitive </rdf:value>
                 <visit.cnt>     1 </visit.cnt>      <interest%>    14 </interest%>
                 <time.in.secs> 15 </time.in.secs>   <time.spent%>  42 </time.spent%>
                 <purchase.cnt>  0 </purchase.cnt>   <purchase%>     0 </purchase%> </rdf:li>
        <rdf:li> <rdf:value> sophisticated </rdf:value>
                 <visit.cnt>     1 </visit.cnt>      <interest%>    14 </interest%>
                 <time.in.secs> 10 </time.in.secs>   <time.spent%>  29 </time.spent%>
                 <purchase.cnt>  1 </purchase.cnt>   <purchase%>   100 </purchase%> </rdf:li>
        </rdf:bag>
     </interested.in>
   </rdf:Description>
</rdf:RDF>
```

Figure 5: An example user profile in RDF produced dynamically by the query given Figure 4

be automatically parsed by user agents and compared with privacy preferences set by the user. Thus, users need not read the privacy policies at every Web site they visit. If a proposal matches the user's preferences, the user agent may accept it automatically by returning a fingerprint of the proposal, called the proposal identifier. If the proposal and preferences are inconsistent, the agent may prompt the user, reject the proposal, send the service an alternative proposal, or ask the service to send another proposal.

The need to query XML documents to extract data is well addressed in the literature, and XML-QL [4] is one of the available query languages. XML-QL has a WHERE-CONSTRUCT clause, like the SELECT-WHERE of SQL, that can express queries, which extract pieces of data from XML documents, as well as transformations, which, for example, can map XML data between DTD's and can integrate XML data from different sources. Although XML-QL shares some functionality with XML's style sheet mechanism, it supports more data-intensive operations, such as joins and aggregates, and has better support for constructing new XML data, which is required by transformations.

3 Functionality and Major Components of the Proposed Architecture

In the proposed architecture, Web browsers are proposed to work in close connection with trusted authorities. When a user logs in a browser as shown in Figure 1, he should provide the URL of the trusted authority, if the user is registered with any. The browser will verify the userid and the password with the trusted authority indicated. If the login is verified, the browser downloads the profile and privacy preferences (possibly in encrypted form) of the user from the trusted authority. If a trusted authority is not specified, the browser is satisfied with its locally available user profiles, if there is any. We propose that browsers should provide required functionality to register a new user either locally, or to any trusted authority the user prefers.

The browser starts a new session for each valid login. During a session, as the user surfs the Web, the click streams are reflected in the user's profile. When the session is terminated with the browser, the updated profile is maintained locally for local user ids. Otherwise, the updated profile is uploaded to the trusted authority (from which it was downloaded).

Note that as one of the current trends in personalization, certain Web servers, for example 'myYahoo', provide an optional registration so that they can serve personalized information. However, since these sites maintain their own userid/password list and user profiles independently, a user is required to obtain different userid/passwords for each site he connects to. In addition, the portions of the user profile are scattered among these sites, because each site has its own interactions with the user. Hence it is not possible to obtain a user complete profile including all the interactions with all the sites he has connected. The trusted authority approach proposed in this paper provides a unique identification and a complete profile for a user at all the sites the user accesses, that is, there is no need to register to these sites individually.

On the server side, the user profile information is exploited in delivering the user personalized content as well as making further recommendations through data mining.

The major components of the system are discussed in three parts: the trusted authority, the client side and the server side.

3.1 The Trusted Authority

The trusted authority is a specialized Web service. It registers users with a userid/password for authentication, and keeps their profiles and privacy preferences information. Users may also provide some personal data that can not be readily obtained from their navigation or purchasing patterns such as their income range, marital status, occupation etc. if they wish to during the registration. Since any Web server may use the user profile, the profile information must have a standard format and well-defined meaning. Therefore, the user profiles and privacy preferences information is stored in XML conforming to RDF as discussed in [2].

The trusted authorities should support at least three main functionality: (1) Register a new user, (2) Provide

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF
     xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#"
     xmlns:rds="http://www.w3.org/TR/WD-rdf-schema#"

  <rdf:Description ID="User">
    <rdf:type resource="http://www.w3.org/TR/WD-rdf
-schema#Class"/>
    <rds:subClassOf resource="http://www.w3.org/TR/
WD-rdf-schema#Resource"/>
    <rds:label>User</rds:label>
    <rds:comment>Users who surfs the Web</rds:comment>
  </rdf:Description>

  <rdf:Description ID="Subject">
    <rdf:type resource="http://www.w3.org/TR/WD-rdf-
schema#Class"/>
    <rds:subClassOf resource="http://www.w3.org/TR/WD-
rdf-schema#Resource"/>
    <rds:label>Subject</rds:label>
    <rds:comment>Subject areas</rds:comment>
  </rdf:Description>

  <rdf:Description ID="interested.in">
    <rdf:type resource="http://www.w3.org/TR/WD-rdf-
syntax#Property"/>
    <rds:label>List of subject that a user has been
interested so far</rds:label>
    <rds:domain resource="#User">
    <rds:range  resource="#Subject">
    <rds:comment>This property indicates the subjects
 that have been visited,
         thus of interest to a user </rds:comment>
  </rdf:Description>

  <rdf:Description ID="total.time.spent">
    <rdf:type resource="http://www.w3.org/TR/WD-rdf-
syntax#Property"/>
    <rds:label>Total Time Spent</rds:label>
    <rds:domain resource="#User">
    <rds:range  resource="http://www.w3.org/TR/WD-rdf
-syntax#Integer">
    <rds:comment>Total time in seconds that has been
actively spent on the Web</rds:comment>
  </rdf:Description>

  <rdf:Description ID="total.no.of.visits">
    <rdf:type resource="http://www.w3.org/TR/WD-rdf-
syntax#Property"/>
    <rds:label>Total Number of Visits</rds:label>
    <rds:domain resource="#User">
    <rds:range  resource="http://www.w3.org/TR/WD-rdf-
syntax#Integer">
    <rds:comment>Total number of sites/subjects that have
been visited</rds:comment>
  </rdf:Description>

  <rdf:Description ID="total.no.of.purchases">
    <rdf:type resource="http://www.w3.org/TR/WD-rdf-syntax
#Property"/>
    <rds:label>Total Number of Purchases</rds:label>
    <rds:domain resource="#User">
    <rds:range  resource="http://www.w3.org/TR/WD-rdf-syntax
#Integer">
    <rds:comment>Total number of items purchased</rds:comment>
  </rdf:Description>
</rdf:RDF>
```

Figure 6: Class and Constraint Property Definitions referenced in the user profile given in Figure 5

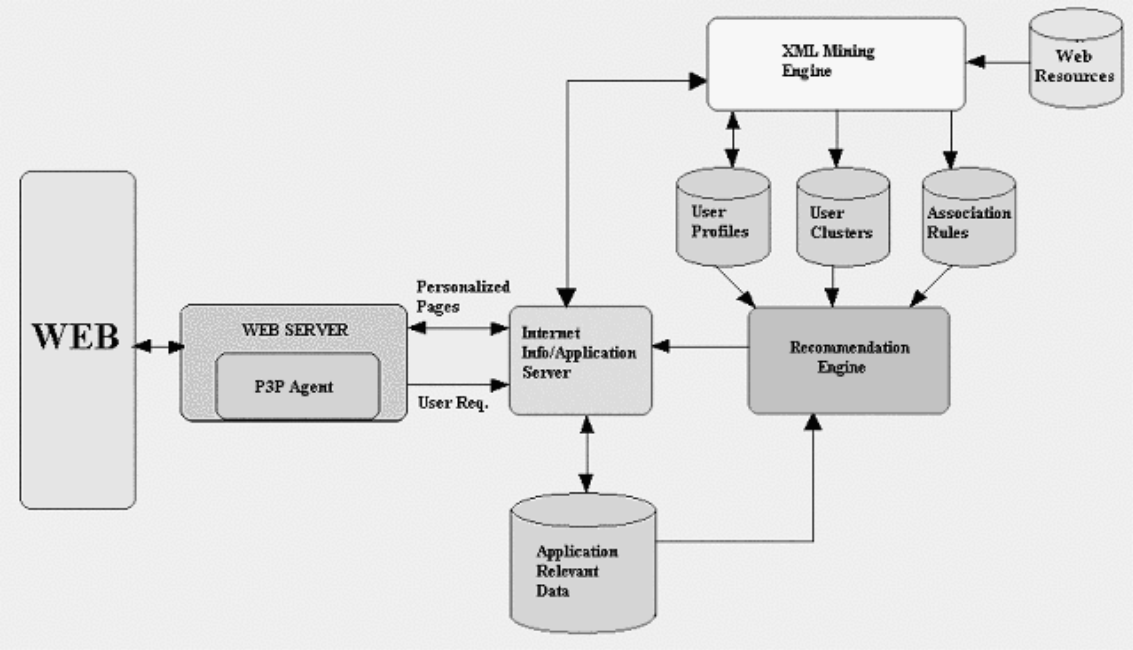

Figure 7: Server Side of the Architecture

user profile and privacy preferences information about an individual user to authorized requests, (3) Update individual users' profiles and preferences with the data sent by authorized updates.

3.2 Client Side of the Architecture

The client side of the system architecture is given in Figure 2. The user, user agent, and the Web browser constitute the client side. In this architecture, the Web browsers together with their user agents are proposed to have the following 'built-in' functionality:

1. 'Login' enforcement,
2. 'Register new user' (remote or local),
3. 'Download user profile' from a trusted authority,
4. 'Upload a user profile' to a trusted authority,
5. Maintain local user profiles,
6. Manage updates to user profiles as a result of new click-streams captured.

When a user initiates a Web browser, the browser asks the user to log in. Each valid login starts a new session. For remote userid/passwords, the previous profile of the user is downloaded from the trusted authority specified (possibly in encrypted form) after the user is authenticated. If no trusted authority is given, the userid/password is assumed to be a locally managed. For locally managed userid/passwords, locally available user profile is used for existing users, and an empty profile is created for new comers.

The user agent captures the navigational history of the user (the click stream) as the user surfs the Web. The user agent, through a proxy object intercepts the Hyper-Text Transfer Protocol (HTTP) stream for observation and alteration. WBI [8] developed at IBM Almaden Research Center, is used for this purpose, and is programmed to act as a proxy object. A proxy receives HTTP requests and forwards them to the appropriate server (referred to as the origin server) for the request to be satisfied. When the origin server returns the results to WBI, they are then forwarded back to the client. Every Web transaction flows through WBI as a request goes from the browser to the Web and the response returns back to the browser.

The navigational history is logged as an XML file whose DTD is given in Figure 3. However since this data is uninterpreted and may be huge, it is not feasible to use it on the Internet for personalization purposes. Therefore descriptive statistics are applied to the user log file by executing a standard XML-QL query shown in Figure 4, which produces a standardized user profile in RDF as shown in Figure 5 that conforms to the schema definition given in Figure 6.

The "User" and "Subject" elements are defined as classes in the schema, and they are subclasses of the 'Resource' class that is enforced by RDF as the top level class. The properties used to describe the interests of a user like "interested.in" or "total.time.spent" are described through "rdfs:range" and "rdfs:domain" instances of RDF's constraint property "rdfs:ConstraintProperty". In describing the resource "interested.in", the range property expresses the fact that the value of the property should be a resource of "Subject" class and the domain property specifies that the property may only be used on resources of "User"

class. This profile is then used to update user's existing profile.

The user agent also handles the privacy preferences of the user to negotiate with the Web servers connected. This negotiation determines how much of the user profile information should be revealed to the servers connected. Note that although this user profile information is valuable, companies or service organizations cannot use it unless a standard mechanism is defined which can programmatically access the information on either the client or the server side. Such a mechanism must also respect a user's privacy constraints. Therefore, a Web site cannot be allowed to retrieve user profile information to which it is not entitled. Sites should not be hindered, however, from accessing the profile information with the informed consent of the user. These issues are addressed in our architecture by conforming to the P3P protocol.

When the user terminates the current session by logging out, the current up-to-date user profile is uploaded to the trusted authority (from where it was downloaded) for remote users.

3.3 Server Side of the Architecture

The server side of the architecture is given in Figure 7. When a user connects to a Web server, the P3P agent of the Web server tries to reach an agreement with the user's User agent. If an agreement is reached, the server obtains the agreed upon portions of the user profile. The user profile information is then used to serve personalized pages and to make personalized recommendations to the user. Since the user profiles are expressed in RDF, any server can process these profiles effectively. Through the use of trusted authorities, the Web servers are relieved from the burden of keeping userid/password registration.

The user profile information can further be processed and used, for example, to form like-minded user groups. An approach in determining the like-minded user groups is to cluster the profiles according to the user interests as well as according to the similarities in total time spent in this resource and total number of visits. In forming the user profile clusters, collaborative filtering techniques can be used. The non-matching subjects of the users in a group are used as a recommendation to the other people in the group. Considering that not only the subjects but also the navigational history is available in the log file, more sophisticated profile groups can be produced. Furthermore as the user navigates through the site, the Recommendation engine can record the navigational behavior of the user and exploit this information in finding association rules that are later used in making site-specific recommendations to the user. Having interoperable user profiles and globally unique user identification, Web servers become capable of sharing server specific data, such as user cluster information, with other Web servers.

4 Conclusions

Web servers need to obtain as much information about their individual users as available to provide them better personalization. Currently, most Web sites providing personalized service require a userid/password registration to distinguish individual users, and they maintain user profiles about individual users on their own. In this way, these profiles can neither be shared with other Web servers, nor they are complete and interoperable. Also, the integration of Internet with telecommunication networks have made it possible for the users to connect to Web with a variety of mobile devices as well as desk tops. This requires that user profiles should be available to any desktop or mobile device on the Internet that users choose to work with.

The 'Trusted Authority' concept introduced in this paper solves the problems mentioned above. Trusted authorities provide a unique userid/password for a user all over the Web, and make the complete user profile available to any authorized device on the Internet, without sacrificing users' rights to privacy. With the help of trusted authorities, Web servers can: (1) be relieved from keeping userid/password information, (2) benefit from all the past interactions of a user with other Web services, and (3) share server information about their users since the users are identified globally by unique authority/userid pairs.

References

[1] A. Buncher and M. Mulvenna, "Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining", in [3].

[2] I. Cingil, A. Dogac, A. Azgin, "A Broader Approach to Personalization", Communications of the ACM, Vol. 43, No. 8, August 2000, pp. 136-141.

[3] A. Dogac, Guest Editor. ACM Sigmod Record Special Section on Electronic Commerce, 27(4), Dec 1998.

[4] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu, "XML-QL: A query language for XML", W3C Document, http://www.w3.org/TR/NOTE-xml-ql.'

[5] P3P Platform for Privacy Preferences Syntax Specification, http://www.w3.org/TR/WDP3P/ syntax.html

[6] Resource Description Framework (RDF) Model and Syntax Specification, W3C Proposed Recommendation. http://www.w3.org/ TR/WD-rdf-syntax.

[7] Resource Description Framework (RDF) Schema Specification, W3C Proposed Recommendation. http://www.w3. org/TR/WD-rdfschema.

[8] Web Browser Intelligence, http://www.almaden. ibm.com/cs/wbi/papers/chi97/wbipaper.html

[9] Extensible Markup Language (XML) 1.0. W3C Recommendation, http:// www.w3.org/TR/REC-xml-19980210.