A Broader Approach to Personalization

Ibrahim Cingil Asuman Dogac Ayca Azgin
Software Research and Development Center
Department of Computer Engineering
Middle East Technical University (METU)
06531 Ankara Turkiye
asuman@srdc.metu.edu.tr

Personalization generally refers to making a Web site more responsive to the unique and individual needs of each user. We propose a broader approach to personalization that provides for interoperability and automation by using the recent data exchange, meta data and privacy standards from the World Wide Web Consortium (W3C), namely, Extensible Markup Language (XML) [12], Resource Description Framework (RDF) [9, 10] and Platform for Privacy Preferences (P3P) [8]. First we briefly discuss these standards.

Extensible Markup Language (XML). XML 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 XML's strengths is its extensibility. Anyone can invent new tags for particular subject areas and they define what they mean in document type definitions (DTDs). 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 inter-operating applications so as to what the tags mean and it is common DTDs which provide for this. A DTD specifies the structure of an XML document by specifying the names of its elements, sub-elements and attributes.

XML-Query Language (XML-QL). The need to query XML documents to extract data is well addressed in the literature and one of the available languages is XML-QL [4]. 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 functionalities 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. There is a need to use recursive functions in certain queries and XML-QL has been extended in this respect in [3].

Resource Description Framework (RDF). RDF 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 an 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.

Platform for Privacy Preferences (P3P). The Platform for Privacy Preferences (P3P) 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.

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 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 System Architecture

Our first aim is to provide dynamically created machine processable user profiles to the Web servers. For this purpose, a user agent at the client side captures the navigational history of the user, that is, user click stream. This navigational history is logged as an XML file which is both human readable and machine processable. However since this data is uninterpreted and 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 on it which produces user profile in RDF. In this way a standard user profile is generated which is, together with its schema, both human readable and machine processable.

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. Such a mechanism must 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 profile information with the informed consent of the user. These issues are addressed in our architecture by conforming to the P3P protocol.

Web servers use the profile information, obtained from the client in conformance with P3P, to deliver personalized information to the clients. The profile information is also useful on the server side to create user profile groups to make suggestions based on the "like minded people" approach. Furthermore, the navigational history of each user on the server site is also kept to determine the site dependent behaviors to make recommendations to like minded users.

Creating a machine understandable user profile at the client side also enables the user agent to start the personalization from the resource discovery, that is, the user profile can be exploited in searching the Internet to find the resources that may be of interest to the user.

The overall view of the system architecture is depicted in Figure 1.

The user log file in XML

There is an agent at the client site, called "user agent" which works in close cooperation with the Web browser to obtain click stream of user actions as s/he surfs the Web. The user agent, through a proxy object intercepts the Hyper-Text Transfer Protocol (HTTP) stream for observation and alteration. WBI [11] 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. This approach uses HTTP which is the fundamental communication mechanism of the Web.

The user click-stream data (i. e., the user log file) obtained this way is kept as an XML document. The log file contains the subjects of resources the user is interested in as well as his/her navigational pattern among the resources. The LOGENTRY element consists of the date, server, request, subject, type, duration and referrer information of the current resource. The DATE element contains the date, time and time zone of the request that can be used in finding out the interests of the user on date basis. The SERVER element holds the server name, IP address and port. The REQUEST element refers to the query/queries the user submits in order to reach the current resource. The ACTION element is the action performed on the current resource like purchase, sell, or view. The SUBJECT element contains the subject of the accessed resource. The TYPE element indicates the type of the current resource, like text, picture, audio. The REFERRER element holds the address, subject and type of the previous resource. This information can be used to produce the user profile groups based on user navigational patterns.

The user agent also records the time spent with that resource. The DURATION element indicates the total time the user has spent with the current resource. Note that we consider the time spent to be indicative only when the user actively interacts with the browser such as moving on the page up and down. If the user is inactive with the browser for a predefined amount of time like a minute then the agent disregards the duration of the inactive connection time.

An entry of the log file produced by the user agent by communicating with the Web browser of the user through the proxy object may be as follows :

```
<LOGENTRY>
    <DATE>
        <DAY> 27 </DAY> <MONTH> 6 </MONTH> <YEAR> 1999 </YEAR>
        <TIME> 16.59 </TIME> <TIMEZONE> +2 </TIMEZONE>
    </DATE>
    <SERVER>
        <NAME> www.metubookstore.com </NAME>
        <IP> 144.122.230.16 </IP> <PORT> 1 </PORT>
    </SERVER>
    <ACTION>  View </ACTION>
    <SUBJECT> Book </SUBJECT>
    <TYPE> Text  </TYPE>
    <DURATION> 2 </DURATION>
</LOGENTRY>
```

The user profile in RDF

The user log file stores log entries indicating the resources that the user surfed but does not provide any interpretation of this data. For this reason we produce a user profile described in RDF obtained by dynamically applying a standard XML-QL query to the user log file. The user profile thus obtained provides interpreted data about the interests of the user. The user profile contains the following information: a list of subjects that the user is interested in; the number of times the user has visited a resource with a specific subject; the cumulative time the user has spent with these resources and the number of items the user has purchased from these resources. The user profile also contains the percent of his interest in this resource which is obtained by taking the percentage of his visit count to this resource to his total number of visits; the percentage of the time he has spent at this resource to the total time accumulated for this user; and the percentage of the count of his purchases from this resource to the total purchase count of the user.

A user profile consists of two parts: A part that is common to all profiles defining the elements mentioned above as the RDF class and constraint property definitions. The second part of the profile contains the user specific values for these elements created dynamically through a standard XML-QL query over the user log file. The details of how user profiles are defined and obtained from the log file are not given here due to space limitations but are available in [2].

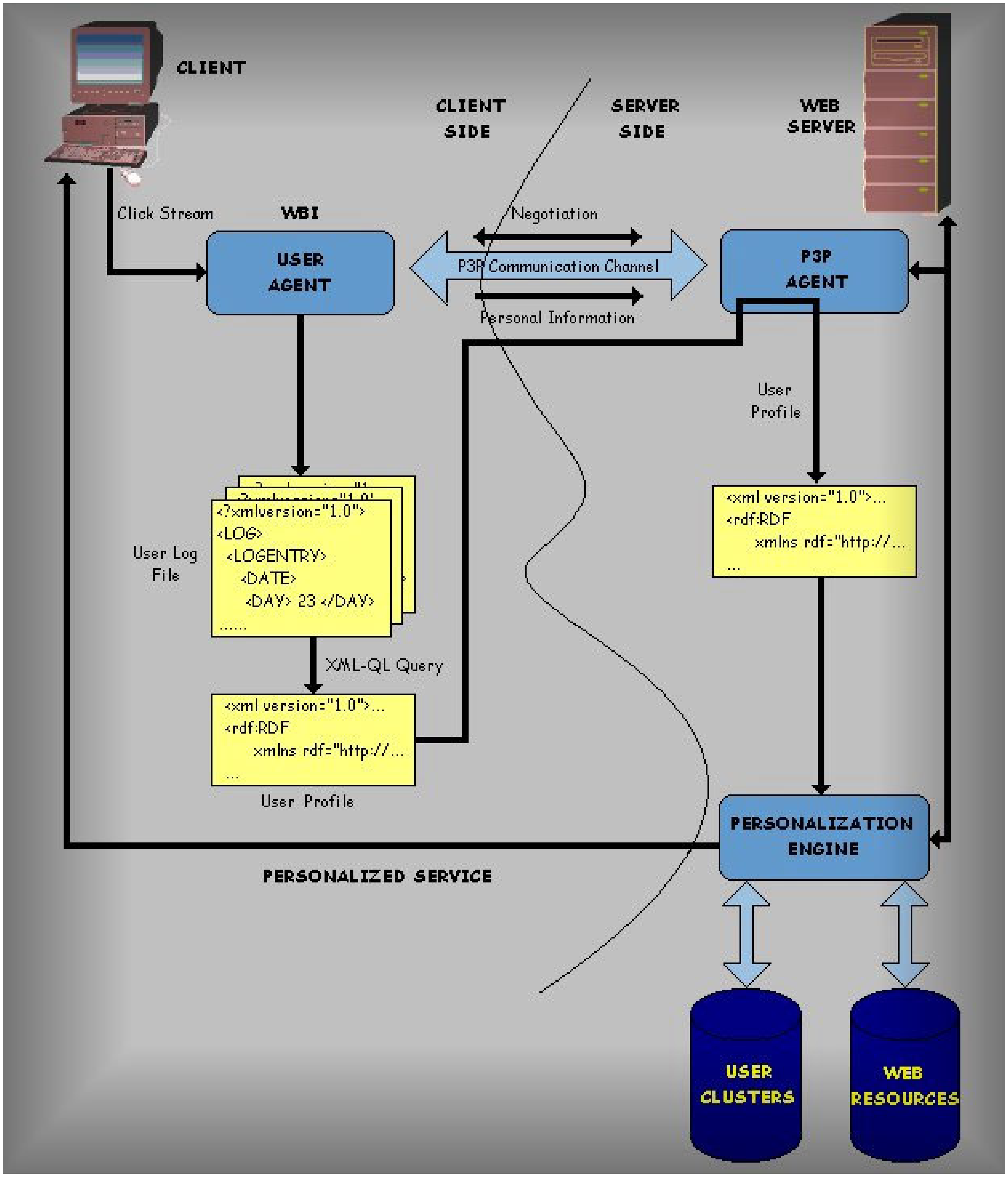


Figure 1. Overview of the system architecture

Exploitation of user profile and the privacy issues

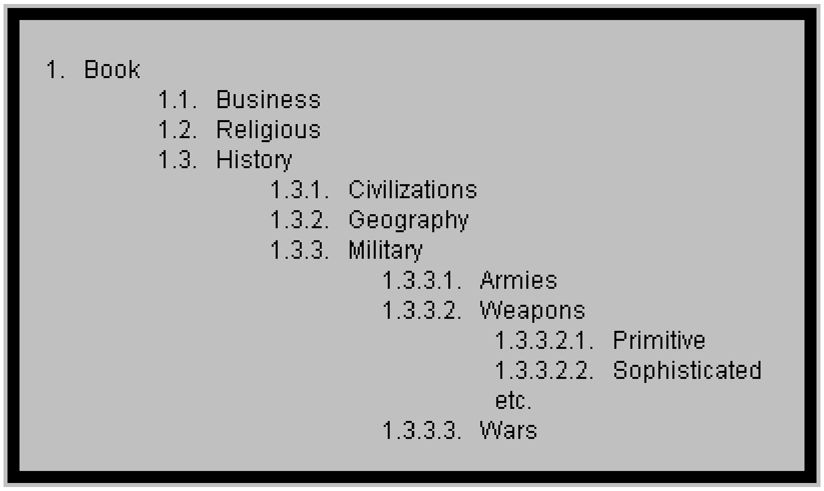
On the server side this profile information is used to deliver the user personalized content, that is, information that fits into his/her personal choices. Moreover, a clustering approach is applied to user profiles to form like minded user groups so that the most likely content or products can be recommended to a user based on his/her similarity to the like minded people and their associated preferences. The navigational history of each user on the server site is also kept to determine the site dependent behaviors to make recommendations to like minded users in this respect too.

The user profile is also useful to discover resources on the Internet that may be of interest to the user as well as obtaining personalized information from these resources. In other words the discovery of the resources are also personalized as opposed to the current practice where personalization starts when a user contacts a Web site. This approach may be preferable in many cases for example when an executive wants to know the information available on the Web that may impact the policies of his company.

As seen, this profile information is valuable, but it must be used in conformance with user's privacy constraints and P3P is used for this purpose. However since most Web servers do not yet support P3P protocol, there is a need to intercept and modify HTTP streams to implement P3P protocol. In our system the user agent, by exploiting the same proxy mechanism used in obtaining the click-stream data of the user, implements the P3P protocol.

For personalized resource discovery meta data tags in HTML documents can be used. However better resource discovery will be possible when resources are expressed in RDF. As stated previously meaning in RDF is expressed through a reference to an application-specific schema. For example Dublin Core (DC) is such an application-specific schema defined by an international resource description community intended to facilitate discovery of electronic resources. Dublin Core [6] defines a schema through its meta data element set which are: Title, Subject, Description, Creator, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, Rights. All of the elements are both optional and repeatable. The values of several elements may be taken from enumerated lists for higher degree of standardization.

For personalization to better capture the user interests, the meta data of the resources should be expressed in finer granularity with a hierarchical structure and with a well-defined subject list to prevent ontology problems. This concept of "concept hierarchies" has also been used in Web usage mining [1]. The concept hierarchy of a book domain may be as given in Figure 2.



1. Book
    - 1.1. Business
    - 1.2. Religious
    - 1.3. History
        - 1.3.1. Civilizations
        - 1.3.2. Geography
        - 1.3.3. Military
            - 1.3.3.1. Armies
            - 1.3.3.2. Weapons
                - 1.3.3.2.1. Primitive
                - 1.3.3.2.2. Sophisticated
                - etc.
            - 1.3.3.3. Wars

Figure 2. The concept hierarchy of the book domain of the example scenario

It should be noted that this meta data description can be specified using RDF class hierarchy and can be very useful in identifying user's interests. For example, when a user reaches the book s/he is looking for directly, for instance by using the search facility of the site, the log file will contain only this entry. In this case the complete sequence of user interest starting from the most generic subject, which is "book", can be obtained from the class hierarchy of the domain. It is important to discover the hierarchical path leading to the desired item since this indicates potential user interest in the items on the path, for example, an interest in history of military weapons also indicates the user interest in "history" and in "military" subjects.

How the approach works

The agent of a user finds out about the resources on the Internet that may be of interest to the user by using "interested in" property in the user profile. The agent starts with a seed resource and finds out new resources by using the resources linked to the seed resource. Then for each resource found it compares the user profile with the RDFs of the resources by matching the user interests with the “subject” element of the resource meta data and selects the ones that match. Note that when the meta data of resources are expressed in Dublin Core, there is no ambiguity on what “subject” means. As the next step the user agent negotiates with the Web service of the resource selected, conforming to the P3P platform. The user agent requests a proposal from the Web service of the resource and compares the proposal and the user's privacy settings in order to decide whether to accept or reject the proposal. If it rejects, the service may send new proposals to the user agent. If an agreement is reached, the user agent stores the address of the resource and the proposal identifier that uniquely describes the agreed proposal with the user agent and the Web service. All the resources that may be of interest to the user and that match the privacy preferences of the user are stored persistently. The user has two choices regarding the discovered resources: s/he may wish to be alerted or the agent may provide the lists of the sites as the user requests.

The profile may contain large amount of data and therefore it is important to be able to query it to extract the information required by the service. Hence by querying the user profile inside the P3P proposal through XML-QL, the service obtains only the related information from the user profile to serve personalized information.

Profile Groups

On the server side, the server creates user profile groups by using the profiles it receives to be able to make suggestions based on the "like minded people" approach.

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 a collaborative filtering technique similar to the one described in [7] is used. In this technique the user profiles are mapped into multidimensional space of concept hierarchy vectors by also taking visit counts into account. After representing each user profile as a vector, the similarities between the profiles are found based on a measure of distance. Based on these similarities user profile clusters are formed. It should be noted that that in [7] this technique is used on user sessions obtained from Web server logs however in our case the technique is applied on the automatically and dynamically obtained user profiles.

The non matching subjects of the users in a group are used as a recommendation to the other people in the group. More sophisticated profile groups can be produced by considering not only the subjects but also the navigational history since this information is also available in the log file.

Once the machine processable data on user profiles is available on the server side, data mining can also be used to extract information from user profiles to offer more specialized services. For example, user may have a seasonal behavior or s/he may be interested in items in promotion.

Conclusions

With mobile electronic commerce seeming to take off, receiving personalized information on the Web will become ever more essential. Yet the techniques used on the Web for personalization purposes should be based on standards to provide for interoperability. The current practices have severe interoperability limitations, for example, cookies from different Web servers have heterogeneities among them and there is no possibility to resolve schematic conflicts generically among different cookie types [1]. Hence a lot of information is being stored with limited use. Another disadvantage is that if the user has never visited a particular Web site before, no historical data is available about the user on that service.

In this article we describe an architecture which overcomes these problems by making use of standards. A log of user's interactions with his/her Web browser is kept in an XML file and a profile of the user that reflects the user's interests and preferences, is automatically and dynamically obtained from this log file. Privacy of the user is preserved through P3P. Any server that a user contacts is able to interpret the user profile and can also maintain like minded user groups again because the information that they receive is highly interoperable. When the meta data of the resources are expressed in RDF it will be a lot easier for agents to discover the resources on the Web that match user profiles. Currently meta data tags of HTML is used for this purpose.

The code implementing the system described is available from http://www.srdc.metu.edu.tr/point.

References

1. Buchner, A. and Mulvenna, M. Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining (in 5).

2. Cingil, I., Dogac, A. and Azgin, A., A Broader Approach to Personalization. Technical Report, SRDC, Middle East Technical University, July 1999, http://www.srdc.metu.edu.tr/publications.html

3. Cingil, I. and Dogac, A. An Architecture for Supply Chain Integration and Automation on the Internet (submitted for publication).

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

5. Dogac, A. Guest Editor. ACM Sigmod Record Special Section on Electronic Commerce. 27(4), December, 1998.

6. Dublin Core. Dublin Core Metadata Element Set. http://purl.org/DC/

7. Mobasher, B., Cooley, R. and Srivastava, J. Automatic Personalization Based on Web Usage Mining, Commun. of the ACM, in this issue.

8. P3P Platform for Privacy Preferences Syntax Specification http://www.w3.org/TR/WD-P3P/

9. Resource Description Framework (RDF) Model and Syntax Specification. W3C Proposed Recommendation. http://www.w3.org/TR/REC-rdf-syntax

10. Resource Description Framework (RDF) Schema Specification. W3C Proposed Recommendation. http://www.w3.org/TR/PR-rdf-schema

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

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