MANAGING PERSONAL DATA IN THE AGE OF CUSTOMIZATION SERVICES

Shi-Cho Cha, eLand Technologies Corporation, csc@eland.com.tw
Yuh-Jzer Joung, National Taiwan University, joung@im.ntu.edu.tw

ABSTRACT

Personalization has become an important feature for information services and applications in recent years. It allows each user to customize his/her preferences (or to be customized) to get the best possible service out of applications. Examples include one-to-one service and CRM (Custom Relationship Management). These services and applications all need mechanisms to manage users' personal data, such as personal profiles, preferences, settings, and their logs. We propose Personal Data Backbone, a mechanism that separates personal data from information services and applications, and provides a globally unique personal folder to store and manage these data. It is not merely a networked storage solution like SAN (Storage Area Network) or NAS (Network Attached Storage), but rather a high-level abstraction for personal data management. The benefits of this mechanism include integration of personal data from various information services and applications, sharing the most up-to-date personal profile (or other personal data), and achieving data integrity. Several critical issues for the design and implementation of Personal Data Backbone are addressed in the paper.

Keywords: Personalized information service, personal data management, mass customization, personal data sharing, integration of personal data.

INTRODUCTION

Personalization has become an important feature for information services and applications in recent years. It allows each user to get the best possible service out of the applications according to his/her preference. Examples include one-to-one services and CRM (Custom Relationship Management).

Personalized services and applications need data about users, such as personal profiles, preferences, settings, and user logs. If every information service and application has its own mechanism for personal data management, people need to maintain their personal data for every service and application. Clearly, it would be very inconvenient for the user to input these data over and over again and to maintain consistency among them.

Moreover, some of the data may be learned by applications via the user's activity in the applications. The more often a user uses an application, the more information about the user the application can collect. When the same user switches to other similar applications, he will most likely exhibit the same behavior pattern and preferences. Obviously, it is very inefficient if these similar applications have to learn the user's behavior separately all over again because, from the user's point of view, quality of service will suffer while switching from a more frequently visited site to another similar but less frequently visited site. Besides, information about use patterns collected from different applications and services can be integrated to become a more complete picture of the person.

We propose Personal Data Backbone, a mechanism that separates personal data from information services and applications, and provides a globally unique personal folder to store and manage personal data.

Separating data from applications is not a new idea. For example, networked storage solution, such as SAN and NAS architecture both adopt this concept to achieve the benefits of cost down, fast to market, simplified storage management (because it is outsourced), higher data availability, and data sharing among clients (6). Compared with Personal Data Backbone, however, SAN and NAS focus more on the low-level management of storages, while Personal Data Backbone focuses more on the management of personal information.

Moreover, when the contents of storage are personal data, some different considerations need to be taken into account. Section 2 describes the concepts and benefits of Personal Data Backbone. Section 3 discusses critical issues concerning system performance, privacy and data integration. Section 4 presents the architecture and components of Personal Data Backbone. Conclusions and future work are offered in Section 5.

PERSONAL DATA BACKBONE AND ITS BENEFITS

Personalized service can be used to improve users' satisfaction and loyalty by reflecting customers' goals, needs and wants (8), as well as to maintain a long term and ongoing interactive relationship with customers.

Personalization is usually expensive in all areas of services, including information services and applications. With the rapid advance of information technologies, however, personalization and mass customization can be easily achieved. For example, recent development in databases enables enterprises to identify and deal with personal data more efficiently. Current interactive technologies make service providers get customers' feedback directly (15). Because of this, more and more enterprises are willing to provide personalization services in order to enhance customer satisfaction and loyalty.

Personalized services and applications need data about users, such as personal profiles, preferences, settings, and user logs. Each personalized service and application has a mechanism for personal data management. We propose Personal Data Backbone to separate personal data from information services and applications, and to provide a globally unique personal folder to store and manage personal data. The benefits of this mechanism include:

First, the module of an application that manages individuals' settings and preferences is separated from the user-independent modules which focuses on the main process and control logical of this application, and is integrated into Personal Data Backbone. Any update to personal data in one application is available to another, and so users need not input their data repeatedly for different applications. For example, one does not need to manually re-bookmark websites he has visited previously when he switches to a new computer. The mechanism will automatically download all requisite information for web surfing when one connects to the Internet, regardless of which computer he is connecting from.

Second, a person may use several applications, among which some information may be shared. For example, if a search engine records a user's search keywords and provides related documents to him, then these history logs may also be used by other news services to filter out unrelated news. In this way users need not maintain personal data with each information service. Furthermore, the history log of each information service is typically incomplete. If these data are collected and integrated, applications can get more complete information about their users so as to provide them better services.

Third, users can also put data objects, such as files or documents, into their personal data folders and gain access to them anywhere and anytime. So when users logon different machines, they can still access their data without manually ftping or remote-copying them to the local sites. From this perspective, Personal Data Backbone can also be viewed as a global file system.

Finally, because logically there is only one such folder for each user, information in the folder is generally the most up-to-date version. Thus, when a user changes some data such as his e-mail address, he does not need to notify his friends but just let them access the latest information in his personal data folder. In this way, even if you haven't contacted your friend for a long time and he has changed his email, you can still get his new email address by accessing his personal folder.

KEY ISSUES

Performance and Scalability

Because of the exponentially increasing Internet population, conventional research in database and file systems is facing a rigorous scalability problem. Maintaining satisfactory performance or even tolerable service quality on such a scale is quite a challenging issue, especially when people use personal information service outdoor via wireless connection, in which performance becomes a critical concern. To solve the performance and scalability problem for Personal Data Backbone, the following problems should be addressed:

First of all, the properties of personal data should be taken into account. For example, in order to provide personalized services, users' activities over the services are usually logged, from which users' interests and behavior patterns can be learned and projected. Such history logs are often updated dynamically. In contrast, some personal data, such as gender, date of birth, and race, are static that do not change over the time. Traditional Internet directory services, such as LDAP (21) and X.500 (9), are designed based on the assumption that the frequency of queries is much higher than that of updates; that is, data is more or less static. They are very suitable to manage the static parts of users' profile, but not to the dynamic part of the profile.

Moreover, replication is often used in distributed systems to improve system performance and scalability. Replication works particularly well for static data. For dynamic data, additional synchronization cost needs to be paid in order to ensure consistency among replicated data sites. In general, the higher the degree of consistency, the higher the synchronization cost, and the lower the degree of scalability. Data consistency is related to data integrity that concerns maintaining correct, complete and timely information about individuals (10). Once again, the degree of data integrity required for applications depends on the type of personal data needed by the applications. Some applications, such as finance and credit checking, require a strong degree of data integrity. So for these applications "lazy updates" schemes for propagating updates to different replica sites may not be appropriate (17).

In addition to replication, data migration is also used in distributed systems to increase system performance (7) (12). Unlike replication, data migration moves data to another site rather than replicates them. To find a good place to migrate the data, the mobility of users should be considered. Migration can be used in tandem with replication. In this case, the system performance depends on how replica data are distributed and migrated. For example, if only the "owner" can update his data objects, then the primary copy of these objects can be placed near to him (and migrates with him) when the "primary copy" replication scheme is used. This again

implies that the type of personal data must be taken into account to achieve optimal performance.

Privacy and Security

In principle, service providers are obliged to protect users' data and their privacy. Service providers should "inform" their users as how their information is going to be shared or manipulated. The user should also be able to "choose" whether to accept these privacy practices or not. Therefore, an important criterion is to make sure that personal data are strictly under their owner's control and that all external accesses to the data should be under the owner's consent.

To enable users to control their privacy preferences, W3C (13) established the Platform for Privacy Preference Project (P3P). P3P is designed for web sites to express their privacy practices to the browser and enable two parties --- the user and the service provider --- to reach an agreement in privacy practices. The practices include declaration of what data are being collected, how they will be used, and whether they are to be shared with other parties (5). The success of such mechanism relies on the trust between the user and the service provider. To ensure that no service provider is abusing the trust, some institute for certification of the service provider is needed. Currently, companies such as TRUSTe (18), CPA WebTrust (4) and Better Business Bureaus, Inc. (3) are established to perform this certification.

With the diverse privacy policy preferences in current use of personal data, we should provide a convenient interface for defining relative privacy policies (16). To ensure a correct and faithful execution of users' privacy policy, the system must deploy a strong security mechanism while providing a user-friendly interface for users to set up their privacy preferences on their personal information.

Note that Personal Data Backbone stores data belonging to individuals. So privacy issues concerned here are between end-users and between end-users and service providers. In contrast, privacy may involve between service providers themselves. This is because service providers may compute and derive information from personal data on their own, and so they should have some control over the data they contribute to the system. Arlein et al. (1) propose the concept of merchants' privacy so that the sharing of personal data and business competence can be balanced.

Data Integration

As mentioned before, Personal Data Backbone deals with different schemas of personal data for different information services and applications. For example, a search engine may predict users' interests by recording the keywords and feedbacks entered by the users. Online shopping sites may also store detail information of individuals' purchases for marketing. If we want to integrate all these data into the personal data management service and do data mining on them, the following two aspects should be taken into consideration.

First, it is important to have a global data schema, into which personal data originating from different sources can be integrated. The definition of such a schema is an interesting future work. Second, if we have different personal data schemas operating between global schema and local personal data or among local personal data themselves, it is useful to have a mechanism to perform data transformation. The problem of integrating data from different sources has been discussed in recent database research, e.g., (14).

SYSTEM COMPONENTS

Personal Data Backbone proposed above consists of many import components. These components are presented in Figure 1. We explain the functionality of the components below:

Personal Data Backbone As Storage for Personal Data

Personal Data Backbone can be viewed as storage for personal data and can be accessed anywhere, anytime, and through any device. So in the backbone there should be a standard interface for requesters to gain access to requested personal data, and a protocol for requesters to exchange messages with the backbone. In this way, LDAP or HTTP can be used for this purpose. However, heterogeneous devices may have different capability. For example, some devices may have a graphic interface for displaying data objects to users, but some may not have this capability. So the interface shall also be flexible. Vanderheiden (19) pointed out that if we wish to achieve this flexibility, information should be stored in a form which is not tied to any particular form of presentation. This means that information is stored in multiple modalities. Users have cross-modality presentation options and can choose in which form to present the information. The system can also provide a text-based auxiliary interface port to allow external hardware or software to query the system and to make a selection from among the available actions.

Although the interface is designed to be flexible so that it can be adapted to different kinds of devices, gateways and translators are still needed in some situations. For example, the backbone may be based on the Internet. When mobile devices are used to access to data in the backbone through, say WAP (20), we need not only extract the data from the backbone, but also translate the data to WAP format (11).

In addition to this, name scheme and data schema are also needed. Naming is important to distinguish data objects and to efficiently locate them. A global name scheme (such as URI (2)) is needed, and the facilities of name and directory services are also needed to allow enquiries about a resource's origin, name and other attributes. Data schema, such as types of personal data and structures of the data, represents the ontology of Personal Data Backbone.

Personal data also need to be integrated in order to achieve the goal of information sharing and synergy mentioned above. For example, if we want to get a more complete picture of a user from his history logs over different applications as opposed to just from the usage patterns of a single application, then a universal data schema is needed. Such schema can integrate heterogeneous data objects into a standard format to be shared among different applications.

A data manipulator takes responsibility for integrating personal data from various sources, converting them into a well-defined format, and performing further manipulations, such as building an index for unstructured documents or multi-language translation.

Personal Data Backbone needs a mechanism to manage data storage and retrieval. As in DBMS and file systems, an individual's personal data should be stored in some data structure defined by the personal data schema described above. It is the kernel of Personal Data Backbone that takes responsibility for personal data management and provides access control to them.

Finally, the overall system should be enclosed by a secure environment. Security mechanisms shall be employed to protect the data. Communication should also be protected by a secure channel. In order to make sure that information about individuals is accessed only under their

agreement, an authentication mechanism is used to verify users' identity. An access control mechanism is used further to check if requesters have the authorization. A digital signature may also be required when the sources of requests or replies need to be proven.

Personal Data Backbone As the Personal Data Provider

When service providers need personal data such as people's interests or behavior patterns for providing customization service, or when merchants wish to get these personal data for marketing (or other purposes), Personal Data Backbone can play the role of personal data provider. Personal Data Backbone not only maintains a person's static profiles (such as his/her birthday, gender, residence, occupation, etc…) but also has *derived data managers* doing the job of generating derived data attributes from personal data. For example, a kind of derived data called "interests of a person" can be generated from his/her behavior logs. Then a derived data manager can be designed and implemented under defined interface for the purpose of transforming a person's logs into interests of him/her. After it is plugged into the system, it maintains interest lists of people. Service providers can then use such information to customize service, and merchants can send advertisements to potential customers. Issues about how to derive data from logs are beyond the scope of this article and will be discussed in our future work.

Of course, privacy shall be guaranteed. People may have their own privacy policies. The privacy agreement negotiator negotiates with requesters of personal data, judges requesters' identities and their purposes, and decides whether the requesters can gain access to the personal data based on individuals' privacy preference settings. If requests are accepted, the privacy agreements about the results of negotiation and their validation period are sent back.

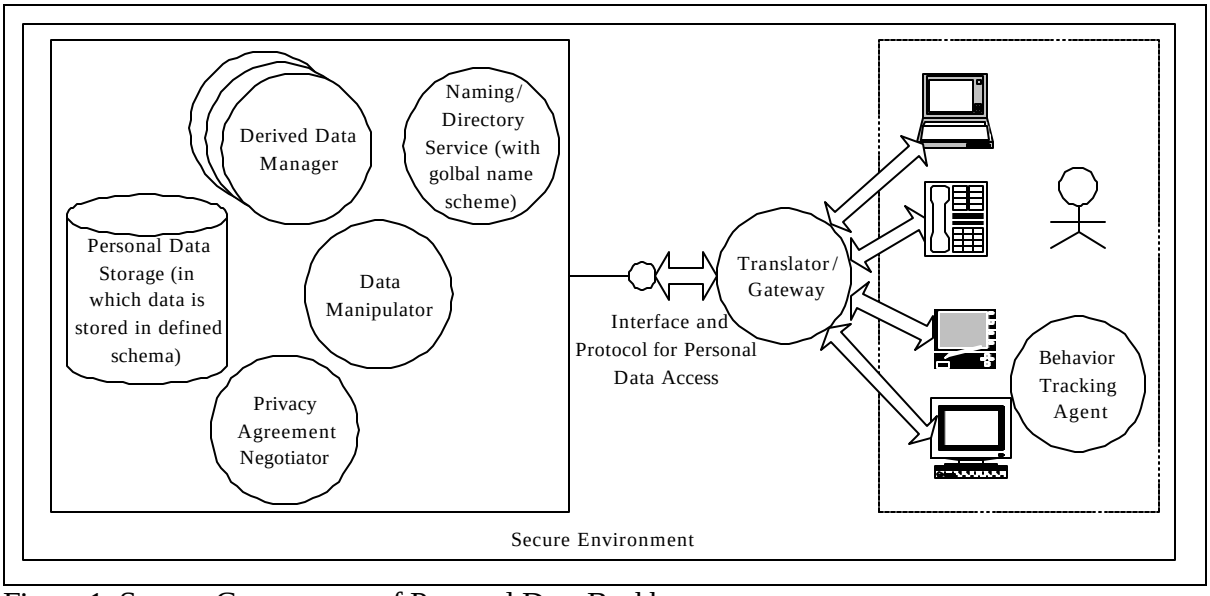

Figure 1. System Components of Personal Data Backbone

CONCLUSION AND FUTURE WORK

The purpose of this article is to propose Personal Data Backbone, a mechanism that separates personal data from information services and applications, and provides a globally unique personal folder to store and manage personal data. The benefits of this mechanism include

integration of personal data from various information services and applications, sharing the most up-to-date personal profile (or other personal data), and achieving data integrity. Our future work will focus on the implementation of this mechanism.

REFERENCES

1. Arlein, R. M. et al. (2000). Privacy-Preserving Global Customization. Proceedings of the 2nd ACM conference on Electronic commerce, 176-184.
2. Berners-Lee, T., & Fielding, R., & Masinter, L. (1998). Uniform Resource Identifiers (URI): Generic Syntax. RFC2396.
3. Better Business Bureaus, Inc. (No Date). Web pages. Retrieved 6 21, 2001 from the World Wide Web: http://www.bbbonline.org/.
4. CPA WebTrust. (No Date). Web pages. Retrieved 6 21, 2001 from the World Wide Web: http://www.cpawebtrust.org.
5. Cranor, L. F. (1998). Putting it together: Internet privacy: a public concern. netWorker, 2(3), 13-18.
6. Gibson, G. A., & Meter, R. V. (2000). Network attached storage architecture. Communications of ACM, 43(11), 37-45.
7. Gavish, B., & Sheng, O. R. L. (1990). Dynamic file migration in distributed computer systems. Communications of ACM, 33(2), 177-189.
8. Hanson, W. (2000). Principles of Internet marketing. South-Western College Publishing.
9. ITU-T (1993). Information Technology – Open Systems Interconnection – The Directory: Overview of Concepts, Models and Services; ITU-T Recommendation X.500 | ISO/IEC 9594-1.
10. Jajodia, S. (1996). Database security and privacy. ACM Computer Surveys, 28(1), 129-131.
11. Kaasinen, E., et al. (2000). Two approaches to bringing Internet services to WAP devices. Proceedings of the 9th World-Wide Web Conference.
12. McCann, J. A., et al. (1995). Dynamic file migration to support parallel database systems. Proceedings of the Third International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 352 -356.
13. W3C. (2000). Platform for Privacy Preference (P3P). Retrieved 6 21, 2001 from the World Wide Web: http://www.w3c.org/P3P/.
14. Papakonstantinou, Y., & Garcia-Molina, H., & Widom, J. (1995). Object exchange across heterogeneous information sources. Proceedings of Intl. Conf. on Data Engineering, 251-260.
15. Peppers, D., & Rogers, M. (1999) The One to One Manager. Doubleday & Company, Incorporated.
16. Reagle, J., & Cranor, L. F. (1999). The platform for privacy preference. Communications of ACM, *42*(2), 48 - 55.
17. Rodriguez, P., & Sibal, S. (2000). SPREAD: Scalable Platform for Reliable and Efficient Automated Distribution. Proceedings of the Ninth World Wide Web Conference.
18. TRUSTe. (No Date). Web pages. Retrieved 6 21, 2001 from the World Wide Web: http://www.truste.org/.
19. Vanderheiden, G. C. (1997). Anywhere, anytime (+anyone) access to the next-generation WWW. Proceedings of the Sixth World Wide Web Conference, 1997.
20. Wireless Application Protocol Forum. (2000). Wireless Application Protocol. Retrieved 6 21, 2001 from the World Wide Web: http://www.wapforum.org/.
21. Wahl, M., & Howes, T., & Kille, S. (1997). Lightweight directory access protocol (v3). RFC2251.