

## ユーザプロフィール管理エージェントの提案と試作

神場 知成, Laurent Freléchoux

NEC C&C 研究所

概要: WWW 上のユーザプロフィール管理エージェントについて述べる。ユーザが WWW 上で情報を効率的に得るためにさまざまなサイト上のサービスがパーソナライゼーション機能(それぞれのユーザごとにカスタマイズした情報を提供すること)を提供している。しかし、パーソナライゼーションを提供するために必要なユーザプロフィール(ユーザ情報)をどこに蓄積するかは難しい問題である。今までは「ユーザプロフィールをサーバ上に蓄積する方法」と「クライアント上に蓄積する方法」の2つがあったが、それぞれ利点と欠点を持っている。ユーザプロフィールをサーバ側に蓄積すると、蓄積されたプロフィールはそのサービスだけで利用可能なものとなり、他のサービスが共有できない。また、クライアント側に蓄積するとサーバ側では複数のユーザプロフィールをまとめて管理していわゆる協調フィルタリングなどのサービスをすることができない。ここで提案するのは、サーバクライアント間のやりとりを仲介するユーザプロフィール管理エージェントを用いて上記の問題を解決することである。ユーザプロフィールはそのエージェント上に蓄積され、サーバが必要とする時には自動的にサーバに送付される。プロフィールの更新も同エージェント上で行われる。筆者等はユーザプロフィール管理エージェントを拡張プロキシサーバとして実装し、複数のサーバ間でのユーザプロフィール共有、複数のユーザ間でのユーザプロフィール共有(グループプロフィールの作成)などを実現した。

## A user profile management agent

Tomonari Kamba and Laurent Freléchoux

C&C Research Laboratories, NEC Corporation

ABSTRACT: This paper describes a user profile management agent on the World Wide Web. To help the web users to get relevant information efficiently, various sites are providing personalized services and the number of such sites is rapidly increasing. However, deciding where to locate user profiles of personalized applications is a difficult problem. To date, there have been two possibilities: storing user profiles on the server, or storing them on the client, which is a terminal that runs a web browser. Both these methods have advantages and disadvantages. When the user profiles are stored on the server, it is impossible for the users to use the profiles to access other personalized servers. When they are stored on the clients, it is difficult for the server to compare different users' profiles and to provide a so-called collaborative filtering service. Our method solves these problems by storing the user profiles on the agent. The agent keeps the user profiles of a group of users and mediates the communications between the users and personalized servers. We have implemented the user profile management agent as an extended proxy server of the web and have demonstrated that the proxy server is useful for sharing user profiles both between different personalized servers and between different users.

## 1. はじめに

最近、パーソナライゼーション (personalization) という言葉を耳にすることが多い。例としてパーソナルニュースがある (英語では personalized news であり personal news ではないが、日本語でパーソナライズドニュースと呼ぶのも不自然なので、ここではこう呼ぶ)。これは大量の情報ソースの中からそれぞれのユーザが興味あるであろう記事を取り出し、わかりやすい形で提示する「個人向け新聞」を作る仕組みである。このようなものが注目されるのは、WWW 上で非常に多種多様な情報がアクセス可能になり、ユーザがその中から役立つものを取り出す仕組みを必要としているからであるだろう。検索エンジンはもちろん有効だが、それは主として手に入れた情報に明確に決まっていればキーワード等を使って検索する場合である。それに対して、ちょうど我々が新聞を読む場合のように毎日コンスタントに有効な (かと言ってあまり特定のキーワードで指定できるわけではない) 情報を手に入れた場合には、WWW 上の情報をフィルタリングしてくれるパーソナライゼーションが有効である。

さて、それではパーソナライゼーションを行うために技術的には何が必要だろうか？まず、パーソナライゼーションエンジンがユーザを知らなければならない (図1)。ユーザに関する知識は通常ユーザプロフィールと呼ばれる。次に、エンジンはユーザプロフィールを参照しながら情報の収集とフィルタリングを行う。最後に、フィルタリングされた情報をユーザに提示する。図1でわかるように、鍵はユーザプロフィールである。ユーザプロフィールにどのような情報をどのようなフォーマットで蓄積するかは当然重要であるが、

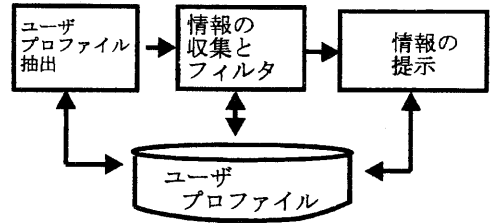


図1. パーソナライゼーションのしくみ  
それはアプリケーションによって異なる。これに対し本論文が主題とするのは、ユーザ情報をどこに蓄積するかである。

パーソナライゼーション機能を提供するために、ユーザプロフィールを WWW サーバ側に蓄積することも、クライアント側、つまり WWW ブラウザが動作している端末上に蓄積することも可能である。次章で見るように、これら2つのアーキテクチャの違いは直接システム機能に影響する。

この論文では、パーソナライゼーションを行う方法として上記いずれとも異なるアーキテクチャを提案する。それはサーバクライアント間の通信を中継し、ユーザプロフィール蓄積や管理を行うユーザプロフィール管理エージェントの利用である。我々は同エージェントを WWW プロキシサーバの拡張として実装した。以下ではその動作や、アプリケーションについて述べる。

## 2. 関連研究および問題点

### 2.1. サーバ上でのユーザプロフィール管理

パーソナライゼーション機能を提供する1つの方法は、ユーザプロフィールをすべてサーバ上で管理するものである (図2)。ユーザからのアクセスがあると、サーバはまず何らかの方法でユーザを認証し、対応するユーザプロフィールをデータベースから取り出し、それを基に情報の収集およびフィルタリング

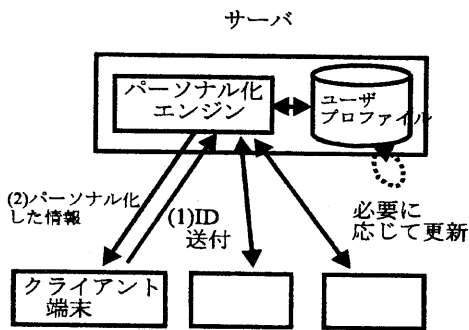


図2. ユーザプロフィールをサーバ側で蓄積、管理する場合

を行い、得られた情報をユーザに提供する。ユーザ認証は通常、ユーザ自ら ID とパスワードを入力することで行う。このようなアプリケーションの例は WWW 上に多数ある。

筆者等の開発した Krakatoa Chronicle<sup>1)</sup>はそのような方法でパーソナライゼーションを行うシステムの1例である。同システムにおいて、ユーザプロフィールはすべてサーバ上に蓄積される。ユーザは WWW ブラウザからサーバにアクセスし、ユーザ ID とパスワードを入力する。システムはそのユーザの興味を特徴づけるキーワードが登録されたユーザプ

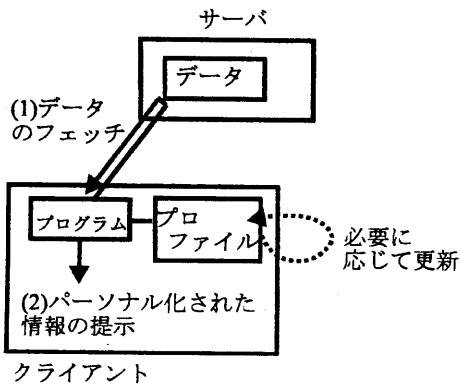


図3. ユーザプロフィールをクライアント側で蓄積、管理する場合

ロファイルを参照し、それを基にそのユーザ向けにパーソナライズした紙面を作成する。

## 2.2. クライアント上でのユーザプロフィール管理

次に、ユーザプロフィールをクライアント側に蓄積する方法を2つ述べる。

- ユーザプロフィールをクライアント側で蓄積、管理する。
- ユーザプロフィールはクライアント側に蓄積されるが、サーバ側が管理する。

前者の方法を利用したシステムのアーキテクチャを図3に示す。この方法でサーバは単なるデータベースとして機能する。クライアント側のプログラムが、クライアント側に蓄積されたユーザプロフィールを参照してサーバにアクセスし、必要な情報を取ってきて画面上に表示する。この方法はたとえば Pointcast Network で採用されている (<http://www.pointcast.com/>)。クライアント上でユーザが読みたいニュース記事のカテゴリを指定すると、それがファイルに記録され、クライアント側プログラムがそれを参照して必要な記事を得る。

2番目の方法を用いた時のアーキテクチャを図4に示す。この方法では、サーバがユーザプロフィールの管理(制御)をするがプロフィールデータ自体はクライアント上に蓄積される。ユーザがクライアント上の WWW ブラウザからサーバにアクセスすると、ユーザプロフィールがクライアントからサーバに自動的に送付され、それに基づいてパーソナライズした情報がクライアントに送付される。ユーザプロフィールを変更する必要がある場合には、サーバがそれをクライアントに通知する。この方法は、Netscape の Cookie<sup>2)</sup>により実装できる。Cookie では、クライアント

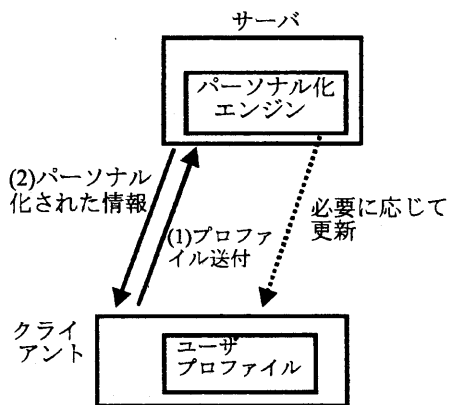


図4. ユーザプロフィールをクライアント上に蓄積し、サーバが管理する場合

上に蓄積される cookie data という特別なファイルを管理するためのサーバクライアント間のプロトコルを規定している。cookie data のファイル中では、一旦ユーザがパーソナライゼーションを行うサーバにアクセスすると、その WWW サーバのアドレスと、ユーザがその次からそのサーバにアクセスする際に送るべき属性と値のペアが記録される。次回以降同じ WWW ブラウザからそのサーバにアクセスをすると、通常の HTTP メッセージに対して、ブラウザが cookie data のファイルを参照して必要な属性および値を自動的に付加した上でサーバに送信する。サーバが、クライアント上に蓄積された cookie-data ファイルを変更したい場合には Set-cookie という文字列を含んだメッセージをクライアントに送付する。

### 2.3. 問題点

既存アーキテクチャの問題点を指摘する前に、パーソナライゼーションの方法として最近注目されているものとして、協調フィルタリングを説明する<sup>3)</sup>。今、あるユーザ X が文書 A に興味があるとすると、システムはその文書 A の内容を解析してユーザ X の興味を抽

出し（たとえばキーワードとして）、その情報を元にそのユーザにとって役立つであろう文書を集めてくることができる。このような手法は通常 content-based フィルタリング(内容に基づくフィルタリング) と呼ばれパーソナライゼーションでよく使われる。協調フィルタリングと呼ばれる方法はこれとは全く異なっている。ユーザ X が文書 A に興味がある時、システムは同じ文書 A に興味を持っている他のユーザを探す。ユーザ Y も文書 A に興味があることがわかると、そのユーザ Y が興味を持っている他の情報をユーザ A にも教える。これが協調フィルタリングであり、システムは文書 A の内容を全く解析することなくユーザ X にとって重要である可能性が高い情報を提示することができる。協調フィルタリングの有効性はすでにいくつかのシステムで実証されている<sup>3)4)</sup>。協調フィルタリングを行うためにはユーザ情報をどこかで一括管理して相互に比較できる必要がある。

以上にに基づきパーソナライゼーションを行ういくつかのアーキテクチャの利点欠点を論じる。

- **ユーザプロフィールの蓄積および管理をサーバ側で行う:** この手法の利点は、content-based フィルタリングと協調フィルタリングの両者を容易にサーバ側で実現できることである。欠点は、クライアント側で全く自由度がなく、同じユーザプロフィールを使って複数のサービスにアクセスする、といったことが不可能なことである。たとえばユーザプロフィールの中にはそのユーザの興味領域などを記入してあるから、異なるサーバが同一内容のユーザプロフィールを利用すると便利な場合があるが、このアーキテクチャではそのようなことができない。

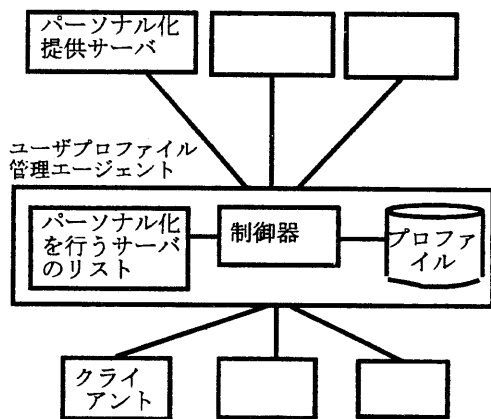


図5. ユーザプロフィール管理エージェント

- **ユーザプロフィールの蓄積および管理をクライアント側で行う:** この手法では、ユーザが自らプロフィールを管理し、複数サイトに同一プロフィールを利用してアクセスすることも可能である。しかし協調フィルタリングのサービスを受けることは難しい。なぜなら、すべてのユーザのプロフィールは各クライアントに分散しており、サーバ側で収集してグループ化するような作業は困難だからである。
- **ユーザプロフィールをクライアント上に蓄積し、サーバが管理する:** この方法でも、上記の場合と同様協調フィルタリングを行うことは実質的に不可能である。

### 3. ユーザプロフィール管理エージェント

#### 3.1. 概要

前章で述べた問題点を解決する手段として、ユーザプロフィール管理エージェントを提案する。これは、グループに属するユーザのプロフィールを蓄積し、それらのユーザとパー

ソナライゼーションを行うサーバとの通信を媒介するものである(図5)。同エージェントが提供する機能を次に示す。

- サーバがユーザプロフィールを要求した時に、要求されたプロフィールを返す。
- パーソナル化された情報をサーバから受け取ると、それを対応するユーザに渡す。
- ユーザが、自分がサーバに送付するメッセージにユーザプロフィールを付加することをエージェントに要求できる。
- ユーザが、パーソナライゼーションを行っていない通常のWWWサーバにアクセスしようとする場合には、クライアントからサーバに送られるメッセージはエージェント上を素通りする。

このユーザプロフィール管理エージェントをユーザ側で管理運用した場合には、そのエージェントのユーザは互いにユーザプロフィールを共有し、たとえば協調フィルタリングサービスをエージェント上で行うことができる。また、このエージェントをサービス提供側で運用した場合には、異なるサーバ上にあるサービス間でユーザプロフィールを共有することができる。

#### 3.2. 拡張プロキシとしての実装

ユーザプロフィール管理エージェントの機能を実証するために、WWWの拡張プロキシとして実装した。これはHTTPプロトコルに基づく通信を中継し、Cookieを利用したパーソナライゼーションサービスとユーザを管理する。サーバがクライアント上にユーザプロフィールをセットしようとする、代わりに拡張プロキシがそのプロフィールを蓄積する。

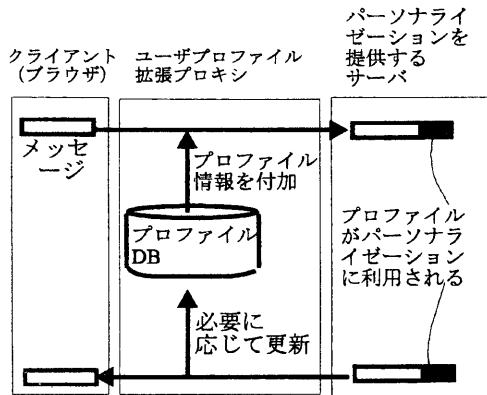


図6. 拡張プロキシを経由した通信

ユーザが自分のユーザプロフィールをサーバに送付する必要がある場合には、クライアントからサーバに送付するメッセージに拡張プロキシが代理でユーザプロフィールを付加する。この拡張プロキシは、たとえば企業等の内部に閉じたイントラネットと、外部に開かれたインターネットとのゲートウェイとして使うことができる。それにより企業内のユーザはお互いにプロフィールを共有して協調フィルタリング等、知識の共有を利用したサービスを受けることができる。

我々は、拡張プロキシをUNIX上でCERNのサーバをベースに実装した(<http://www.w3.org/pub/WWW/Daemon>)。パーソナライゼーションを行わないアドレス(URL)に対するアクセスに関しては通常のプロキシサーバとして機能する。クライアントからHTTPプロトコルによって要求があると同プロキシはまず、要求されたURLと要求したユーザのユーザIDとをデータベースで探し、それがパーソナライゼーションを行うサービスに関するものであるかどうかを調べる。そしてそれが登録されていれば、対応するユーザプロフィールを取り出してメッセージにCookieのフォーマットで付加してサーバに送

付する(図6)。

サーバ側でユーザプロフィールを変更した場合には、それを通常のHTTPに基づくレスポンスに付加する。レスポンスはプロキシを経由してクライアントに送付されるので、その過程でプロキシがユーザプロフィールを取り出しデータベースを変更する。一旦ユーザプロフィールを取り出してしまうと、プロキシはメッセージからユーザプロフィールに関する部分の記述を取り除いてクライアントに渡す。

パーソナライゼーションを提供するサーバにユーザが初めてアクセスする場合には、サーバはHTTPレスポンスにユーザプロフィールを含めて返し、それによってサーバがパーソナライゼーションを提供していることがプロキシに通知される。プロキシはそのサーバのアドレスと、得られたユーザプロフィールをデータベースに追加する。このような拡張プロキシにより以下のような利点がある。

- サービスプロバイダはユーザプロフィール管理の必要がなく、コスト削減ができる。
- サーバ側ではユーザプロフィール管理に伴うデータベースとのインタフェースなどを考慮する必要がなくなる。
- ユーザプロフィール蓄積が拡張プロキシ上に集中化されるので、異なるWWWサービス間でユーザプロフィールを共有したり、ユーザ間でプロフィールを共有できる。
- ユーザ認証を1ヶ所に集めることができるので単純化できる
- 同じプロキシを利用すれば、別のクライアントからパーソナライゼーションを利用したサーバにアクセスする場合でも、同じユーザプロフィールが利用できる。
- ブラウザから見て完全に透明である。

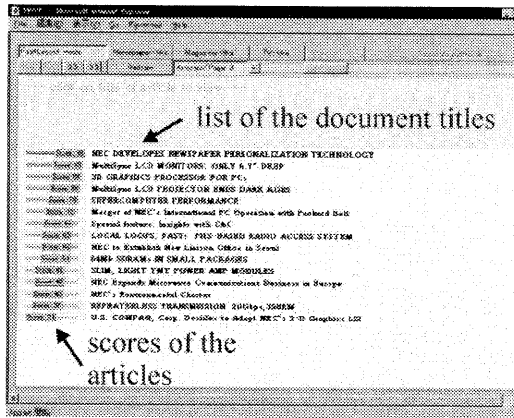


図7. パーソナルニュース（または技術レポート）の画面例

### 3.3. アプリケーション

拡張プロキシ上で異なるサービスがユーザプロフィールを共有する例として、パーソナルニュースとパーソナル技術レポートを作成した。これら2つは異なるWWWサーバ上にあり異なるコンテンツを提供するが、パーソナライゼーションに関しては同じエンジンを利用している。つまり、それぞれのユーザの興味傾向を示す「キーワードと重み」を利用してパーソナライゼーションを行うサービスである。あるキーワードに関連する文書にユーザが興味を持っているとき、そのユーザのプロファイルにおいてそのキーワードは高い重みが割り当てられている。このサービスを行うサーバは、ドキュメントを外部サイトから集めてキーワードを抽出し、それをサーバ内に蓄積する。ユーザがサーバにアクセスすると、システムは各ドキュメントのキーワードとユーザプロフィールを比較し、すべてのドキュメントにスコアづけをする。ユーザプロフィールの中で高い重みを持つキーワードが多く含まれるドキュメントほど高いスコアを得る。高いスコアを得たドキュメントは、タイトルリストの中で上位に表示される（図7）。ユーザプロフィールは拡張プロキシ上

にCookieのフォーマットで蓄積される。

従来のようにCookieファイルがクライアント上に蓄積される場合には、異なるサーバ上のWWWサービスが同じユーザプロフィールを共有することはできなかったが、本拡張プロキシを利用することで可能になる。たとえば、パーソナルニュースはよく読んでいるがパーソナル技術レポートを読んだことがないユーザが初めて技術レポートを見た時にでも、そのユーザがパーソナルニュース上でたとえばインターネットに関する記事をよく読んでいるのであればユーザプロフィールがそのように設定されているので、興味ある技術レポートを容易に見つけることができる。

ユーザプロフィール拡張プロキシの第2の利用法として、複数のユーザプロフィールを拡張プロキシ上で共有するサービスを開発した。これは協調フィルタリングの1例である。あるユーザが拡張プロキシを経由してパーソナライゼーションサービスにアクセスすると、通常であればそのユーザのプロファイルだけに基づいてパーソナライゼーションが行われるが、協調フィルタリングを利用すると他のユーザプロフィールも利用する。たとえば、システムがある記事のスコアをつける場合に、そのユーザだけでなく他のユーザのプロファイルも考慮する。これは、たとえば興味ある技術レポートを読む時に、自分の興味（判断）だけでなく、同じグループに属する他の人の興味も参考にすることに相当する。

ユーザプロフィールがサーバ上に蓄積されている場合には、ユーザ側でそれを利用できないので、サーバ側が協調フィルタリング機能を用意している場合にしかそのようなサービスを楽しむことができないが、本拡張プロキシを利用すればユーザ側が主体的に協調フィルタリングサービスを実現可能である。

#### 4. 議論と今後の課題

##### 4.1. セキュリティとプライバシー

パーソナライゼーションは、プライバシーの問題と密接に関係している。WWW 上でパーソナライゼーションサービスを利用する場合には、何らかの方法で個人情報を盗まれる危険性がある。我々の拡張プロキシを利用した場合にはユーザ情報はプロキシ上に蓄積されることになるが、プロキシは通常ファイアウォールなどとして外部から進入できない位置に置かれているので、少なくとも外部サーバ上にユーザプロフィールを蓄積する場合と比較すればプライバシーの問題は少ない。

我々の拡張プロキシにユーザがアクセスする際もユーザIDおよびパスワードを要求されるが、それはあくまでパーソナライゼーションを行うサーバにアクセスしようとした場合だけであり、要求されるのは同じブラウザを開いている限り最初の1度だけである。また、一旦プロキシに入ってしまうと、そこからWWWサーバへのアクセスは `anonymous` であり、サーバ側からユーザを特定できないメリットもある。

##### 4.2. 今後の課題

これまでのところ本拡張プロキシ上で実現したのは比較的簡単なアプリケーションである。さらに進んだサービスとして、ここで示したようなグループプロフィールの利用だけではない協調フィルタリングの実現などを計画している。同じプロキシのユーザを自動的に分類し、同じグループに属する人がより強くユーザプロフィールを共有できるしくみなども実現していきたい。また、拡張プロキシはユーザプロフィール管理に限らず他の多くの付加機能を組み込んでいくことが可能である。ユーザの知識を蓄積して高度なサービスを提供するような「知的プロキシ」についても検討をしていきたい。

##### 5. おわりに

現在のユーザプロフィール管理方法の問題点を指摘し、それを解決する方法としてパーソナライゼーション提供サーバとクライアントとの通信を中継するユーザプロフィール管理エージェントを提案した。同エージェントはユーザプロフィールを蓄積、管理する。それをWWWの拡張プロキシとして実装した。この拡張プロキシを利用することで、異なったサーバ間でユーザのプロフィールを共有したり、異なったユーザ間でプロフィールを共有したりすることが可能になった。

パーソナライゼーションはユーザが役立つ情報を効率的に得るための重要な技術であるが、プライバシー等にも関連するため慎重に実装するべきものである。我々のユーザプロフィール管理エージェントにより、サービス提供者が容易にパーソナライゼーションサービスを管理できるようになり、ユーザが協調フィルタリングなどのサービスを受けるためにも役立つと考えている。

##### 参考文献

- 1) Kamba, T., et al., Krakatoa Chronicle - An Interactive Personalized Newspaper on the Web, Proc. of the Fourth Int. Conf. on the WWW, pp. 159-170 (1995).
- 2) 石川：知られざるクッキーの謎, Internet Magazine 1996.11, pp.258-263 (1996).
- 3) Hill, W., Stead, et al., Recommending and Evaluating Choices in a Virtual Community of Use, CHI'95 Conf. Proc, pp. 194-201 (1995).
- 4) Shardanand, U. and Maes, P., Social Information Filtering, CHI'95 Conf. Proc, pp. 210-217 (1995).