

Web サービスにおけるパーソナルな情報の共有と蓄積 Personal Information-Sharing and Accumulation in Web Services

岡野真也† 小岩秀和† 後藤文太郎‡
Shinya OKANO Hidekazu KOIWA Fumitaro GOTO

1. はじめに

Web2.0 アプリケーションの利用は、従来「こちら側」に置いていたデータを「あちら側」へ置くことになる。すなわち、従来は PC 等にローカルに保存していたデータを、Web アプリケーション側に保存する^[1,2]。

それらの Web2.0 アプリケーションでは、Web API の公開等によりサービス連携を可能にしているものも多いが、柔軟なデータの利用やセキュリティの面での問題があると考えられる。

たとえば、Google が提供しているウェブメールサービス Gmail^[3]、カレンダーサービス Google Calendar^[4]がある。現在、Gmail では 2.8GB 以上の大容量のストレージを使用することができる。さらに、Google の検索技術を搭載した高度なメール管理機能を備えている。そして、Google Data APIs^[5]等を利用することで、それらのアプリケーションの連携も可能となっている。

一見すると利用者にとってとてもメリットがあるように思えるが、コンテキスト型広告配信等において、利用者のメールが処理されている^[6]といった点等、セキュリティ面に注意しなくてはならない。

データの管理主体はサービス提供側となってしまっているため、ユーザが自分自身のデータを自由かつ柔軟に管理・利用することもできない。

そこで、本研究では、Web2.0 アプリケーションにおいても、ユーザが自分自身のデータを自由かつ柔軟に管理・利用できる「個人情報空間」と呼ぶパーソナルな情報を蓄積・共有するためのフレームワークを提案する。

2. 既存 Web サービス vs. 個人情報空間

現在いくつかの Web サービスが公開され、いくつかのマッシュアップアプリケーションが開発されている。しかし、これらの Web サービスはインターフェースしか提供しておらず、ユーザは自身のデータがどのように利用、処理されているかを知り、管理することができない。

Google の Gmail では API が公開されていて、アプリケーションから利用することができる。しかし、Gmail に置いたユーザのデータは、Google の利益のために利用される。また、いつどのように使われたのかをユーザはまったく知ることができない(図1)。

上記の問題を解決するために、個人情報空間フレームワークでは、データはユーザ側に置いて、ユーザ自身が管理をできるようにする。アプリケーションがインターフェースとなる Web サービスを利用するのは従来と変わらないが、ユーザデータへのアクセスに個人情報空間フレームワークの API を利用する(図2)。

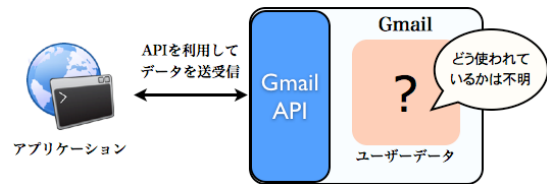


図1 既存の Web サービスの問題 (Gmail)

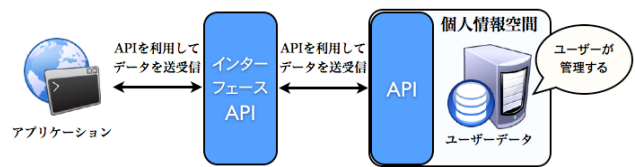


図2 個人情報空間のモデル

3. 個人情報空間フレームワーク

個人情報空間を実現するためのフレームワークにおけるデータの取扱について説明する。

メール、カレンダーなどは時間とコンテンツを組み合わせたデータであり、これらは一つのデータベーステーブルで扱えることが分かる。同様にアドレス帳、ブックマークなどは場所に名前を付けたデータであり、これらもテーブルを統一することができる。このようにテーブルを統一することで、時間情報や場所情報といった抽象的なデータの扱いができるようになる。

ユーザデータベースとデータのやり取りをする、時間や場所のような抽象的なデータを扱う API と、メール、カレンダー、アドレス帳、ブックマークのような具体的なインターフェース API を Web サービスで開発した。図3に個人情報空間フレームワークの概要を示す。

API は WSDL にて公開され、通信は SOAP で行う。データベースには、Java と相性の良い MySQL を使用した。

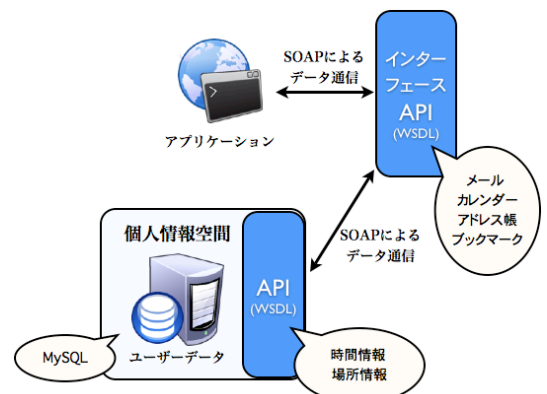


図3 個人情報空間フレームワークの概要

†(株)エストコスモ, EST COSMO Co.,Ltd

‡北見工業大学, Kitami Institute of Technology

4. 個人情報空間フレームワークの利用例

開発した個人情報空間フレームワークを用いて、メールやカレンダーなどのインターフェースを単独で利用したアプリケーションや、メールやブックマークを組み合わせた複合的なアプリケーションを開発することができる。

4.1 ブックマークアプリケーションの例

ブックマークアプリケーションの一覧表示の例を用いて、処理の手順を述べる。

(1) IDリストの取得

ブックマークのアプリケーションは、サーブレットからサービスを利用している。まず、BookmarkServlet は、BookmarkInterface サービスのデータベースアドレス、ユーザ名を引数にして getBookmarkList を呼ぶ。getBookmarkList の内部では、DataType=Bookmark を引数に追加して AddressAliasAccess サービスの getAddressAliasList を呼ぶ。AddressAliasAccess サービスでは、JDBC を使って SQL を実行して結果セットを得た後、アイテムの ID リストを getBookmarkList に返す。getBookmarkList は受け取った ID リストを BookmarkServlet に返す。これで BookmarkServlet はブックマークの ID リストを取得する (図4)。

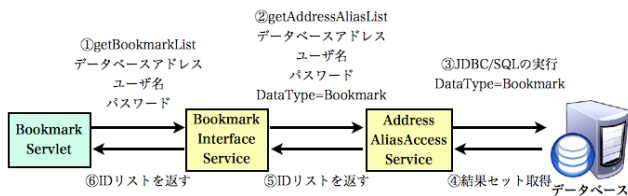


図4 IDリストの取得

(2) データの取得

次に、BookmarkServlet は(1)で得た ID とデータベースアドレス、ユーザ名、パスワードを引数にして BookmarkInterface の getBookmark を呼ぶ。getBookmark の内部では、AddressAliasAccess の getAddressAlias を呼ぶ。getAddressAlias では、JDBC を使って SQL を実行し、結果セットを得た後に AddressAlias のデータを getBookmark に返す。getBookmark は受け取った AddressAlias のデータから Bookmark データを抽出して BookmarkServlet に返す。これで BookmarkServlet はブックマークのデータを取得できる (図5)。

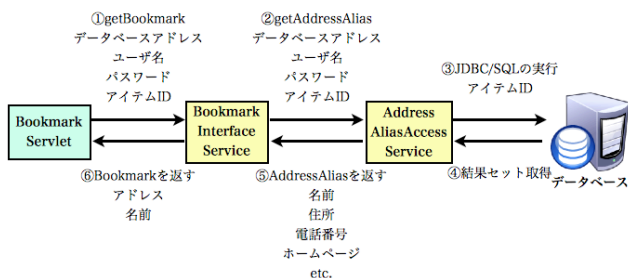


図5 データの取得

(3) 表示

取得したデータを HTML に加工して表示する。

4.2 ソーシャルブックマークの例

アドレス帳とブックマークを組み合わせた、ソーシャルブックマークのようなアプリケーションを図6に示す。このアプリケーションでは、最初に、登録された複数ユーザの個人情報空間のアドレスをアドレス帳のインターフェースを利用して得て、その個人情報空間のアドレスにアクセスする。次に、ブックマークのインターフェースを利用して各ユーザのブックマークを収集して、それらをまとめた一覧の作成を行う。重複する URL は Rank としてカウントする。

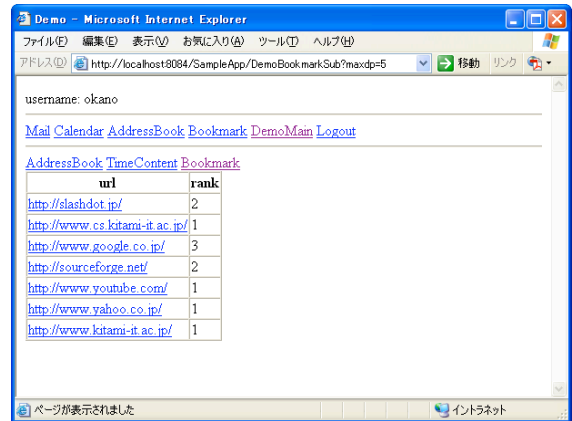


図6 アドレス帳とブックマークの組み合わせ

また、これらのインターフェースを利用して、さらに新しいインターフェースを開発することも可能である。

5. まとめ

現在の Web サービスの問題として、ユーザデータをユーザ自身が管理できないことを述べた。

この問題の解決方法として、ユーザ自身がデータを管理し、API を通して各種インターフェースがデータを利用する個人情報空間フレームワークを提案した。個人情報空間フレームワークの開発を行い、その利用例を示した。

今後の課題としては、個々のデータへのアクセス権限などのセキュリティの問題などを挙げる事ができる。

参考文献

[1] 梅田望夫：ウェブ進化論，筑摩書房(2006).
 [2] 野村直之：データの置き場所は「あちら側」？「こちら側」？，
<http://itpro.nikkeibp.co.jp/article/Watcher/20060921/248714/>
 [3] Google， <http://mail.google.com/>
 [4] Google， <http://calendar.google.com/>
 [5] Google， <http://code.google.com/apis/gdata/>
 [6] Google：Gmail プライバシーポリシー，
<http://www.google.com/mail/help/intl/ja/privacy.html>