

ユーザプロフィール情報を管理するための リソース指向データレポジトリの検討

川西 直^{†1} 宮森 良昌^{†1} 大橋 正良^{†1}

きめ細やかで質の高い「あなただけ」のコンテキスト適応型サービスを実現するためには、より多様なプロフィール情報を、より多くのサービス提供者に対して開示することが可能な機構が必要である。このような考えに基づき我々は、ネットワーク中に分散して遍在する情報源から多種多様なプロフィール情報を収集して格納し、一元的に管理することのできるデータレポジトリ「HoRUS (History of Resources in Ubiquitous Services)」の検討を行っている。サービス提供者に対してこのデータレポジトリへのアクセス権を与えることによって、格納された多種多様なプロフィール情報を開示することが可能となる。本稿では、データレポジトリの要求要件と設計指針を述べるとともに、基本機能を確認するために行った試作について述べる。

A Resource Oriented Data Repository to Management User Profiles

NAO KAWANISHI,^{†1} YOSHIMASA MIYAMORI^{†1}
and MASAYOSHI OHASHI^{†1}

For realizing context-aware services especially customized for you, it is necessary to construct a mechanism that enables to show various profile information to various service providers. From this perspective, we are researching a data repository called HoRUS (History of Resources in Ubiquitous Services) that collect and store various profile information from many sources. By allowing for service providers to access this data repository, a user can show his/her profile information to them. In this paper, we describe the requirements and the design policy of the data repository, and the prototype implementation to verify its fundamental functions.

1. はじめに

「いまだけ・ここだけ・あなただけ」のフレーズで体現されるようなきめ細やかなコンテキスト適応型サービスを提供するためには、サービス提供者側がユーザに関連する「いまだけ・ここだけ・あなただけ」の情報、いわゆるコンテキスト情報を入手することが必要不可欠である。「いつでも・どこでも・だれでも」というフレーズで体現されるネットワーク環境は現実のものとなりつつある今、実空間と仮想空間とにまたがって形成されるネットワークは、多種多様で膨大な数のサービスであふれかえっている。このような膨大な数のサービスが「いつでも・どこでも・だれでも」利用できてしまう環境においては、ユーザにとって「いまだけ・ここだけ・あなただけ」の適切なサービスを選択することが、ユーザ側からもサービス提供者側からも最重要課題である。これに向けて、コンテキスト情報を「ヒント」にユーザとサービスとの適切なマッチメイキングを行うコンテキスト適応型サービスへの期待が高まっている。

現状のコンテキスト適応型サービスでは、コンテキスト情報とサービス提供者とが直結しているモデルが多く見受けられる。例えば、これまでに研究開発が進められてきたコンテキスト適応型サービスでは、コンテキスト情報を抽出する技術に焦点が置かれていることが多く、コンテキスト適応型サービスは抽出されたコンテキスト情報を直接利用することを前提に垂直統合的に構築されている。また、GPS 搭載携帯電話の普及によって日常的に利用するコンテキスト適応型サービスの代表例となった位置情報に応じたサービスにおいては、位置情報を取得する機能こそサービスとは切り離されたコンポーネントとして提供されているが、取得した位置情報を直接サービスに渡す形で構築されている。あるいは、コンテキスト情報としてユーザ自身によるサービス利用履歴を用い、ユーザ本人は他のユーザの利用履歴を比較することでより質の高いサービスを提供しようとするサービスにおいては、コンテキスト情報である利用履歴そのものがサービスによってのみ管理されている場合もある。

一方、モバイルデバイスや情報家電の増加、あるいはインターネット上で提供されるサービスの増加に伴い、ユーザに関する情報を生み出す情報源が非常に多くなっており、それらの情報源は分散して遍在している現状がある。たとえば、ユーザが持ち歩く携帯電話にはカメラや GPS、加速度センサなどの多様なセンシング機能が搭載されており、ユーザの「いまだけ・ここだけ・あなただけ」を示すコンテキスト情報の情報源として活用すること

^{†1} 株式会社国際電気通信基礎技術研究所メディア情報科学研究所
Media Information Science Laboratories, Advanced Telecommunications Research Institute In-

ternational

が可能である。また、携帯電話機のストレージの拡大に伴い、カメラで撮影した画像や動画、これまでに送受信したメールの情報、通話履歴、Web 閲覧履歴などは、ユーザの「あなただけ」を示す情報として非常に有用であり、コンテキスト適応型サービスへの活用も考えられる。情報家電が普及した家庭内には、ユーザが選択したコンテンツが録画された HDD レコーダや、ユーザがデジタルカメラで撮影した写真を蓄積したフォトリージなど、ユーザに関連付けられた膨大な情報が存在する。さらにはインターネット上にはブログ、ソーシャルネットワークサービス (SNS)、オンラインフォトアルバムサービスや動画共有サービスなど、やはりユーザに関連付けられた膨大な情報が存在している。これらの情報はユーザに対して「あなただけ」のサービスを提供する上で非常に有益な情報であり、かつユーザ自身にとってもこれらの情報をコントロールしてこそ「わたしだけ」のサービスを享受することができる。

本研究は、「いまだけ・ここだけ・あなただけ」のコンテキスト適応型サービスにおいて、特に「あなただけ」のサービスの提供するためのコンテキスト情報の取り扱いに焦点をあてた研究である。以下本稿では、「あなただけ」のサービスのために必要となる、ユーザ自身に関連付けられたコンテキスト情報を「プロフィール情報」と呼ぶこととする。

今後、よりきめ細やかで質の高い「あなただけ」のコンテキスト適応型サービスを実現するためには、より多様なプロフィール情報を、より多くのサービス提供者に対して開示することが重要であると考えられる。開示するプロフィール情報の種類を増やすことにより、サービス提供者はユーザをより詳しく把握することが可能となる。また、プロフィール情報を開示するサービス提供者を増やすことにより、複数のサービス提供者が提供可能なより多様なサービスの候補から適切なサービスを選択することが可能となる。そのためには、プロフィール情報とサービス提供者との密な相互接続を切り離し、遍在している多様なプロフィール情報を容易に組み込むことができるようにするとともに、複数のサービス提供者に対して容易に開示することのできる機構が必要不可欠である。

このような考えに基づき我々は、分散して遍在する情報源に存在する多種多様なプロフィール情報を収集し、一元的に管理することのできるデータレポジトリ「HoRUS (History of Resources in Ubiquitous Services)」の構築を行っている。HoRUS はユーザ毎に構築されるデータレポジトリであり、ユーザに関連するプロフィール情報をユーザ毎に個別に構築されるものである。HoRUS は、遍在する情報源に存在するプロフィール情報を収集して、の中から特定のユーザに関連する情報のみを収集し、ユーザの「プロフィール情報」として再編成するデータレポジトリであり、コンテキスト適応型サービスのサービス提供者は、ユー

ザ毎に構築された HoRUS を参照することによって、ユーザのプロファイル情報を入手可能となり、ユーザに対してコンテキスト適応型サービスを提供することができる。

本稿では、まず 2 で HoRUS を活用することで実現されるコンテキスト適応型サービスのサービスシナリオについて述べる。次に 3 では、2 で述べたシナリオを実現するための HoRUS の要求要件を洗い出すとともに、HoRUS の設計指針について述べる。その後、4 では 3 に基づいて実装を行った HoRUS のプロトタイプについて述べる。最後に 5 でまとめとする。

2. サービスシナリオ

ここでは、ユーザに関するプロフィール情報を収集して一元的に管理するデータレポジトリである HoRUS を用いることによって、従来のようにプロフィール情報をサービス提供者とが密に相互接続したモデルでは実現し得ない、きめ細やかな「あなただけ」のコンテキスト適応型サービスが提供されることを、いくつかのサービスシナリオを通じて述べる。

2.1 センシング情報を複数のサービス提供者に開示するシナリオ

ユーザ A は携帯電話に搭載された GPS で取得した位置情報を HoRUS に登録して管理している。A は、位置情報に応じて飲食店を推薦してくれる複数のサービス提供者 (X と Y) に対して位置情報を開示し、両者から推薦される飲食店の中から気になる飲食店を利用することにした。この際、A は X と Y に対して自身の HoRUS へのアクセス情報を教えるとともに、自身の HoRUS に対して X と Y からのアクセスに対して位置情報のみを開示するようにアクセス許可を設定した。これにより X と Y は、A の HoRUS にアクセスすることによって A の位置情報を取得できるようになり、取得した位置情報に応じて飲食店を推薦し、メールなどの Push 型のシステムを用いて推薦情報を配信する。このようにして、A は X と Y から配信された飲食店の中から気に入った店を選択して利用することができるようになった。(図 1 参照)

2.2 サービス利用履歴を異なるサービス提供者間で共有するシナリオ

オンラインの書籍販売サイト I の利用者であるユーザ B は、I で様々な書籍を購入している。B は I における自身の書籍購入履歴を参照し、同じ情報を自分の HoRUS へと蓄積して管理している。

ある日 B は、利用者購入した書籍の情報を利用し、同じ書籍を買った他のユーザの情報と照らし合わせることによって、別の書籍を推薦するなどのサービスが話題の書籍販売サイト J の存在を知った。B は早速 J を利用してお薦めの書籍を推薦してもらおうと思ったが、

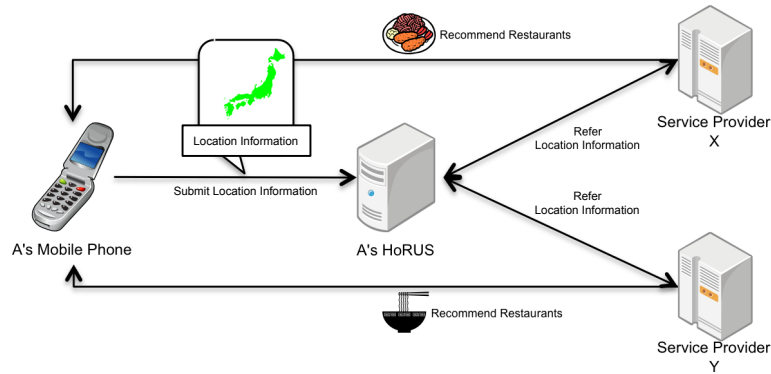


図 1 センシング情報を複数のサービス提供者に開示するシナリオ

J を初めて利用する B は J に書籍の購入履歴がなく、適切な推薦サービスを楽しむことができない。そこで B は、自身の HoRUS で管理している I で購入した書籍の履歴を J に対して開示することにした。B は J に対して自身の HoRUS へのアクセス情報を教えるとともに、自身の HoRUS に対して J からのアクセスに対して I での購入履歴のみを開示するようにアクセス許可を設定した。これにより J は、B の HoRUS にアクセスすることによって B の I における購入履歴を取得できるようになり、その情報をあたかも J における購入履歴かのように用いて推薦アルゴリズムを処理することによって、B に対してお薦めの書籍を推薦することが可能になった。(図 2 参照)

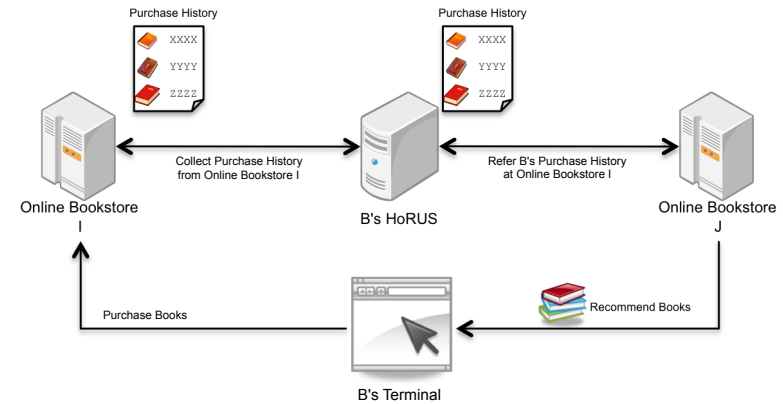


図 2 サービス利用履歴を異なるサービス提供者間で共有するシナリオ

2.3 プロファイル情報を用いたアクセス制御シナリオ

ユーザ C は、デジタルカメラで撮影した写真を HoRUS にアップロードして管理している。ある日 C は友人 D の家へ遊びに行き、D の家にある大型テレビに自身の HoRUS にアップロードしてある写真を表示しようと思った。そこで C は、自身の HoRUS で管理している写真の中から表示したい写真を選んで D の家のテレビに対して開示することにした。C は D の家のテレビに対して自身の HoRUS へのアクセス情報を教えるとともに、自身の HoRUS に対して D の家のテレビからのアクセスに対して特定の写真のみを開示するようにアクセス許可を設定した。これにより、D の家のテレビに C の HoRUS 上で管理している写真を表示することができた。

また、HoRUS に設定したアクセス許可には、C が D の家の近くにいるとき、という条件を与えていた。C は自身の位置情報を定期的に HoRUS に登録しており、この位置情報を

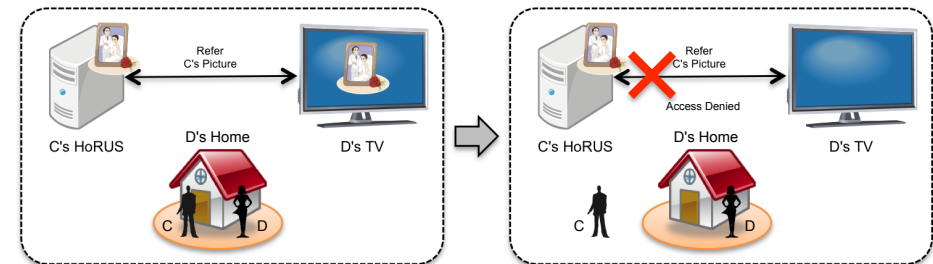


図 3 プロファイル情報を用いたアクセス制御シナリオ

活用することで、C の位置情報が D の家から遠ざかった場合に D の家のテレビに対するアクセス許可を剥奪することができる。これにより、C が D の家から帰った後、D の家のテレビは C の HoRUS 上の写真に対してアクセスすることができなくなった。(図 3 参照)

2.4 プロファイル情報の見える化によるライフログシナリオ

ユーザ E は、日常生活で取得可能な様々なプロフィール情報を自身の HoRUS に登録して管理している。プロフィール情報としては、オンラインショッピングサイトの購入履歴、オンラインアルバムサイトにアップロードした写真情報、自身の位置情報、加速度センサで取得した運動情報、家庭内における電力消費量など、様々な情報を蓄積している。E はこれ

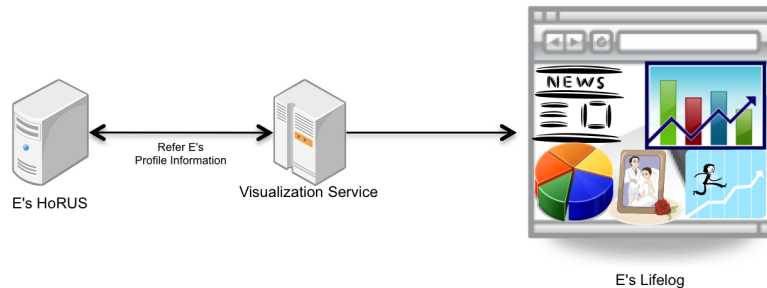


図4 プロファイル情報の見える化によるライフログシナリオ

らの情報を種々のサービスに対して開示してコンテキスト適応型サービスを楽しむ一方で、プロフィール情報見える化サービスに対して開示することによって、プロフィール情報を統計的に処理し、視覚的にわかりやすく表示させることで、日々の行動を振り返るためのライフログとして利用している。(図4参照)

3. 設 計

ここでは、2で述べたサービスシナリオを実現するためにHoRUSに求められる要求要件について整理するとともに、それらの要求要件を満たすための具体的な設計指針について述べる。

3.1 要求要件

2で述べたような多様なサービスシナリオを実現可能なHoRUSを構築するためには、以下のような要求要件が挙げられると考えられる。

3.1.1 多様なプロフィール情報を格納可能であること

例として提示したシナリオだけでも、ユーザの位置情報、書籍販売サイトでの購入履歴、写真データそのものなど、実に様々な種類のプロフィール情報が格納されている。また、位置情報や写真データなどの例ではデータそのものがユーザによってHoRUSに登録される一方、書籍販売サイトの購入履歴の例では、本来書籍販売サイトIにある購入履歴を参照してHoRUS上で扱っている。このように、必ずしも全てのプロフィール情報がHoRUS上に直接適に格納されるわけではなく、外部の様々なサービスと連携して運用可能であることも求められる。

本研究で構築を目指す個人のプロフィール情報を一元的に管理するデータレポジトリに

おいては、特定のプロフィール情報の格納だけを目指すのではなく、これら多種多様なプロフィール情報を格納可能なデータレポジトリとして構築することが求められる。ここで、世の中で使われ得るプロフィール情報を完全に列挙して、それらを格納可能な形でデータレポジトリを構築するアプローチも考えられるが、プロフィール情報に限らずコンテキスト情報は、サービスを楽しむために必要なあらゆる情報と考えられており¹⁾、他のサービスにとっては何ら有益な情報に見えない情報であっても、特定のサービスにおいてはその情報を解析することで非常に質の高いコンテキスト適応型サービスを提供することが可能であるケースも考えられる。このような観点から、プロフィール情報をあらかじめ体系付けるのではなく、できるだけ簡単に多様なプロフィール情報を格納できるような仕組みであることが求められる。

3.1.2 多様なプロフィール情報源からプロフィール情報を収集可能であること

格納すべき多様なプロフィール情報は、ユーザの携帯電話やホームサーバ、インターネット上のWebサービス、あるいは周囲に存在するセンサネットワークなど、多様なプロフィール情報源に遍在しており、データレポジトリに格納するためには、これらの情報源からプロフィール情報を収集しなければならない。特にユーザの周囲に存在するセンサネットワークなどは、ユーザの移動に伴って参照すべき情報源が変化することなども考えられる。このように多様かつ動的に変化する情報源からプロフィール情報の収集を可能にするためには、データレポジトリへプロフィール情報を格納するための基本となるインタフェースやプロトコルを定義することが重要である。その上で、プロフィール情報の種類に応じてインタフェースやプロトコルに対する拡張を施すことが可能であることが求められる。

3.1.3 多様なサービス提供者に対してプロフィール情報を開示可能であること

例として提示したシナリオにおいて、HoRUSに格納されたユーザのプロフィール情報は、飲食店推薦サービス、書籍販売サイト、友人の家にあるテレビ、など様々なサービス提供者に対して開示されている。また、それぞれのサービス提供者に対して開示するプロフィール情報の種類を切り替えることも必要である。つまり、本研究で構築を目指すデータレポジトリには、多様なサービス提供者からの呼び出しに応じて適切なプロフィール情報を開示するための機構が必要である。

3.1.4 プロフィール情報に対するアクセス権を柔軟に管理可能であること

HoRUSに格納されたプロフィール情報は、当然のごとくユーザに関する様々な情報が含まれているため、不特定多数に対して自由なアクセスを許可することは望ましくない。例として提示したシナリオにおいては、位置情報のみをXとYにのみ公開する、書籍販売サイ

ト I の購入履歴のみを J にのみ公開する、あるいは D の家のテレビに対して C が D の家の近くにいる間だけ公開する、などプロフィール情報の種類に応じて、またプロフィール情報を開示する相手に応じて、さらにはユーザ自身のプロフィール情報に応じて、アクセスの可否を柔軟に制御できることが望ましい。また、HoRUS 上のプロフィール情報へのアクセスするサービスも多様であることから、多様なサービス提供者の参入を前提とした管理機構であることが望ましい。

3.2 設計指針

ここでは、これまでに述べてきたような要求要件を持つデータレポジトリである HoRUS に関して、全ての要件を満たすための設計指針について述べる。具体的には、WWW 上のリソースに対する統一的なインタフェースの重要性を説いた REST アーキテクチャスタイル²⁾に着目する。様々なプロフィール情報源から収集したプロフィール情報を、HoRUS 上でリソースとして取り扱うことのできる形で格納するとともに、サービス提供者に対してもプロフィール情報をリソースとして開示可能なデータレポジトリとして設計する指針について述べる。

3.2.1 REST アーキテクチャスタイル

REST (Representational State Transfer) アーキテクチャスタイルとは、分散システムにおいてコンポーネントのスケラビリティと独立性を最大化しつつ、通信量と遅延を最小化するようなネットワークソフトウェアのアーキテクチャのスタイルであり、WWW の構成要素に対する統一的なインタフェースの重要性を説いたものである。

REST では、名前を付けることができるあらゆる情報を“リソース”と呼び、統一的なインタフェースを用いてリソースを制御することの重要性を説いている。具体的には、すべてのリソースは URI により管理され、URI で識別されたリソースに対して HTTP の GET, POST, PUT, DELETE の 4 つのメソッドを許すことで、リソースの状態の取得、リソースの作成、リソースの状態の変更 (更新)、リソースの削除を行う。このように REST では、WWW のあらゆる構成要素をリソースという単位で取り扱い、それに対する 4 つのシンプルな操作のみを認めることで、リソースの高い再利用性を実現している。

昨今 WWW 上で展開される種々のサービスにおいて、マッシュアップアプリケーションの成功などから各サービスの持つ情報の再利用性が強く認識され、情報を公開して他のサービスとの間で共有するための手法として、REST アーキテクチャスタイルをベースにした手法への注目が高まっている。具体的には、REST に沿ったアーキテクチャであるリソース指向アーキテクチャ (ROA) や RESTful インタフェースなどの形で、REST をベースに

したインタフェースの定義などが盛んに行われている³⁾。

3.2.2 リソース指向のプロファイルデータレポジトリ

本研究では、プロフィール情報を多様なサービス提供者に対して開示する際には、プロフィール情報の再利用性の高さが重要になると考え、統一的なインタフェースを用いることによってリソースの再利用性を実現している REST に着目し、プロフィール情報のデータレポジトリをリソース指向に構築する。具体的には、全てのプロフィール情報をリソースとして扱い、全てのリソースに URL を付与し、HTTP の 4 つのメソッドを用いての制御を許可する。

3.1 で挙げた要求要件のうちの、「多様なプロフィール情報源からプロフィール情報を収集可能であること」という要求要件に対しては、HTTP の POST を用いてデータレポジトリ上にプロフィール情報のリソースを生成できるようにする。HTTP のライブラリは各種プログラミング言語で広く提供されているため、多様なプロフィール情報源からプロフィール情報を収集するために、データレポジトリへのプロフィール情報生成機能を情報源に組み込む場合にも、あるいは情報源と独立して実現する場合にも、容易に実装することが可能である。

「多様なサービス提供者に対してプロフィール情報を開示可能であること」という要求要件に対しては、プロフィール情報のリソース毎に付与された URL に対して HTTP の GET を用いることで、プロフィール情報を取得することができるようにする。先にも述べたような HTTP のライブラリの普及率の高さに加え、コンテキスト適応型サービスにおいて WWW 上の RESTful インタフェースを用いて情報を再利用することも有用であることも多く、WWW 上の情報を再利用するようにプロフィール情報を再利用することのできるこのアプローチは優位であると考えられる。

「多様なプロフィール情報を格納可能であること」という要求要件に対しては、URL によって識別されるリソースを HTTP の 4 つのメソッドを用いて制御するというリソース指向の考え方においては、リソースの持つ意味については言及されていないため、プロフィール情報が特定のセマンティクスに制約されることなく、多様なプロフィール情報を取り扱うことができる。

最後に「プロフィール情報に対するアクセス権を柔軟に管理可能であること」という要求要件に対して、全ての通信におけるステートレス性の重要性を訴える REST においては常に認証情報が含まれており、その情報を元にアクセス権を付与することは可能であると考えられる。一方で実際の WWW 上ではサービスのパーソナライズ化を目的として、様々なセッ

ション情報が実装されているが、分散環境におけるオープンな認証基盤である OpenID⁴⁾ とオープンな認可基盤である OAuth⁵⁾ が普及しつつある中、これらを用いたセッションの取り扱いの統一⁶⁾ も提案されている。HoRUS においても同様のセッションの取り扱いを導入した上で、アプリケーションレベルでアクセス権の柔軟な管理を実現できると考えている。

4. 試作実装

我々はこれまでに述べてきたようなサービスシナリオ、要求要件、および設計指針に基づき、個人プロフィールデータレポジトリである HoRUS の試作を行った。試作では、HoRUS の基本機能を確認することを目的とし、3.1 で挙げた要求要件を満たす機能のいくつかを実装した上で、実際にプロフィール情報を収集して格納するとともに、プロフィール情報を開示することで享受可能なサービスの試作も行った。

4.1 個人プロフィールデータレポジトリの試作

今回の試作では、HoRUS の基本機能のうち、要求要件における「多様なプロフィール情報を格納可能であること」、「多様なプロフィール情報源からプロフィール情報を収集可能であること」、および「多様なサービス提供者に対してプロフィール情報を開示可能であること」、の3点について確認することを目的とし、Apache-Tomcat を利用した Web アプリケーションとして実装を行った。

「多様なプロフィール情報を格納可能であること」という要求要件に対し、格納可能なプロフィール情報の種類をプラグイン形式で拡張できるように実装を行った。XML で記述されたプラグインには、拡張した格納可能にするプロフィール情報の名前、およびプロフィール情報に含まれる項目の名前とその型が記載されている。GPS で取得した位置情報を格納するためのプラグインを図5に示す。この場合、プラグインの名前が「gps_location」で、「observed」(観測時刻)を日時表記で、「latitude」(緯度)および「longitude」(経度)を正規表現にマッチした文字列で、「height」(高度)を整数値で、含むことができるようになっている。このようなプラグインを組み込むことによって、HoRUS は内部でデータを管理するデータベース上にこのプロフィール情報を格納するためのテーブルを作成する。なお、内部でデータを管理するデータベースとして今回の試作では MySQL を使用した。

「多様なプロフィール情報源からプロフィール情報を収集可能であること」を満たすために、Atom Publishing Protocol⁷⁾ (AtomPub) を用いてプロフィール情報を格納することのできるインタフェースを実装した。HoRUS では組み込まれたプラグイン毎に、プラグインの名前に応じて個別の URL が用意される。図5のプラグインを組み込んだ場合は、

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  GPS-Locationプラグイン設定
-->
<plugin name="gps_location" base="plugin" storage="horus.plugin.DBProfileStorage">
  <profile location="gps_location">
    <entity name="observed">
      <content type="datetime" size="32" column="observed" index="false" />
      <format required="true" type="date" option="yyyy-MM-dd'T'HH:mm:ssz" />
    </entity>
    <entity name="latitude">
      <content type="string" column="latitude" size="32" index="false" />
      <format required="true" type="string" />
      <rule method="regex" params="(0|[0-9]+)([1-9][0-9]*|\.[0-9]+)?$" />
    </entity>
    <entity name="longitude">
      <content type="string" column="longitude" size="32" index="false" />
      <format required="true" type="string" />
      <rule method="regex" params="(0|[0-9]+)([1-9][0-9]*|\.[0-9]+)?$" />
    </entity>
    <entity name="height">
      <content type="integer" column="height" index="false" />
      <format required="false" type="integer" />
      <rule method="min" params="0" />
    </entity>
  </profile>
  <actions>
  </actions>
</plugin>
```

図5 GPS で取得した位置情報を格納するためのプラグイン

```
<?xml version="1.0" encoding="utf-8" ?>
<feed xmlns="http://www.w3c.org/2005/Atom">
  <title>gps_location</title>
  <id>http://horus.u-type.net/river24/gps_location/</id>
  <entry>
    <title>6757</title>
    <id>http://horus.u-type.net/river24/gps_location/id/6757</id>
    <link rel="alternate" type="application/atom+xml"
      href="http://horus.u-type.net/river24/gps_location/id/6757" />
    <published>2009-03-30T13:54:36+09:00</published>
    <updated>2009-03-30T13:54:36+09:00</updated>
    <author><name>river24</name></author>
    <content type="application/xml">
      <observed>2009-03-09T00:07:40+09:00</observed>
      <latitude>34.74395</latitude>
      <longitude>135.766333</longitude>
      <height>130</height>
    </content>
  </entry>
  <entry>
    <title>6756</title>
    <id>http://horus.u-type.net/river24/gps_location/id/6756</id>
    <link rel="alternate" type="application/atom+xml"
      href="http://horus.u-type.net/river24/gps_location/id/6756" />
    <published>2009-03-30T13:54:36+09:00</published>
    <updated>2009-03-30T13:54:36+09:00</updated>
    <author><name>river24</name></author>
    <content type="application/xml">
      <observed>2009-03-09T00:07:28+09:00</observed>
      <latitude>34.74415</latitude>
      <longitude>135.766333</longitude>
      <height>128</height>
    </content>
  </entry>
  ....
</feed>
```

図6 GPS で取得した位置情報のプロフィール

http://horus.u-type.net/river24/gps_location のような URL が用意される。この URL に対して対応するプロフィール情報を POST することで、HoRUS 内にプロフィール情報のリソースが生成される。プロフィール情報を表現する書式としては Atom Syndication Format⁸⁾ の Atom Entry の書式を用い、プロフィール情報の種類に応じて必要なタグを拡張して表現している。生成されたプロフィール情報のリソースには、http://horus.u-type.net/river24/gps_location/id/6757 のような URL が付与される。

このようにして生成された URL に対して HTTP の GET をすることによって、格納されたプロフィール情報を参照することが可能になる。また、先ほど POST 時に指定した URL に対して HTTP の GET をした場合は、図6に示すように、直近の10件のプロフィール情報の Atom Entry の一覧を Atom Feed 形式で取得することができる。これらの URL をサービス提供者に通知し参照させることによって、「多様なサービス提供者に対してプロフィール情報を開示可能であること」という要求要件を実現している。

HoRUS に対して様々な情報源から多様なプロフィール情報を収集して格納できることを確認するために、いくつかのプロフィール情報収集プログラムと、そこから収集されるプロフィール情報を格納するためのプラグインを作成した。プロフィール情報としては、GPS デバイスで取得した位置情報、パソコン上で使用しているアプリケーション名や打鍵数、クリック数などのパソコン操作情報、赤外線リモコン信号受信デバイスで受信した信号を解析して取得した家電操作情報、などを検討し、これらのプロフィール情報を取得するプログラ

ムを、HoRUSへプロファイル情報をPOSTする機能を組み込んで実装し、HoRUSにプロファイル情報を格納させた。

また、このようにしてHoRUSに格納されたプロファイル情報を活用するサービスの一例として、ユーザの位置情報の履歴を参照して地図上に表示するWebサービス、家電操作情報とWeb上のテレビ番組表情報をマッシュアップしてユーザが視聴していたテレビ番組名を取得するサービスなどを実装した。

4.2 課題

今回の試作では基本機能の一部を確認するだけの初期的な実装であったため、いくつかの課題が残されている。3.1で挙げた要求要件のうち、「プロファイル情報に対するアクセス権を柔軟に管理可能であること」という要求要件に対する機能が具現化されておらず、プロファイル情報のURLを知ることさえできれば誰でもプロファイル情報を参照できる状態となっている。また今回の試作では、プロファイル情報の種類毎に設定されたURLをサービス提供者に通知して参照させるモデルになっているため、複数の種類のプロファイル情報が必要とするサービスに対しては複数のURLを通知しなければならない。

これらの課題を解決するために、OAuthにおけるリソース開示の流れをベースにした図7のような流れで動作するHoRUSの設計を始めている。

- (1) ユーザはサービス提供者に対し、特定のプロファイル情報のURLではなく、`http://horus.u-type.net/river24/`のような自身のHoRUSのURLのみを通知する。
- (2) ユーザのHoRUSのURLを取得したサービス提供者は、ユーザのHoRUSサーバのURLにサービスに必要なプロファイル情報の種類をクエリパラメータとして含め、HTTPのリダイレクトを使用してユーザを自身のHoRUSサーバへと転送する。ここで、プロファイル情報の種類を自由に拡張可能なHoRUSにおいてはプロファイル情報の明示的な種類の指定が難しいため、プロファイル情報の保持するセマンティクスのようなもので一般化したクエリパラメータを用いることを検討している。
- (3) ユーザは自身のHoRUS上のインタフェースから、サービス提供者から要求されたプロファイル情報を開示するか否か、開示する場合はそのプロファイル情報の中でも具体的にどのプロファイル情報を開示するのか、さらには開示する際の条件を指定し、OAuthにおけるAccess TokenのようなTokenを発行する。
- (4) HTTPのリダイレクトを用いて、発行したTokenをサービス提供者に伝える。
- (5) サービス提供者は伝えられたTokenを用いてHoRUSにアクセスし、プロファイル

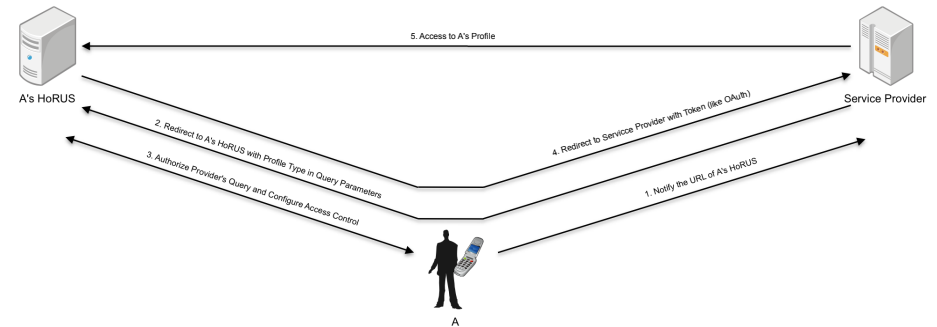


図7 プロファイル情報開示までの流れ

情報を取得する。

この流れを実現するためには、HoRUS上に格納するプロファイル情報の保持するセマンティクスとサービス提供者から指定されたクエリパラメータとのマッチングが必要であり、オントロジなどの導入も検討している。また、あらゆるサービス提供者に対して開示条件をコントロールするのは大変であり、開示条件の再利用やユーザ間での共有など、コントロールコストを下げるメカニズムも重要となるであろう。さらには、OAuthのモデルでは、サービス提供者とHoRUSとの間で事前にリクエストキーの取得が必要であり、サービス提供者の数が膨大で、ユーザ毎に構築されるHoRUSの数も膨大な本研究へ適用した場合、スケラビリティについて吟味する必要がある。

5. おわりに

本稿では、きめ細やかで質の高い「あなただけ」のコンテキスト適応型サービスの実現に向けて、分散して遍在する情報源に存在する多種多様なプロファイル情報を収集し、一元的に管理することのできるデータレポジトリ「HoRUS (History of Resources in Ubiquitous Services)」について、その要求要件と設計指針を述べるとともに、基本機能を確認するために行った試作について述べた。

すでに多様なプロファイル情報が遍在するWWW上には、本研究と関連する技術やサービスも多く見受けられる。WWW上に遍在するプロファイル情報を収集して統合するライフログサービス⁹⁾や、SNS上の情報を用いたアプリケーションを構築するための共通基盤¹⁰⁾があり、オープンな認証基盤を目指すOpenIDにはユーザのプロファイル情報を取得する

メカニズムも存在する。これらの技術やサービスは WWW 上で取り扱うことのできるプロファイル情報の取り扱いに注力しているが、コンテキスト適応型サービス向けに WWW 上の様々なプロファイル情報を収集する取り組み¹¹⁾も存在し、WWW 向けのサービスとコンテキスト適応型サービスの垣根は今後ますます低くなっていくものと思われる。

本研究も同じような考えに基づき、今後はこれらの技術やサービスなどとの差分や親和性について検討を行うとともに、4.2 で述べた課題に対する解決策の具体化を進め、全ての機能を統合した HoRUS の設計と実装を行っていく。

謝辞 本研究の一部は、総務省「ユビキタスサービスプラットフォーム技術」研究開発の成果である。

参 考 文 献

- 1) A. K. Dey, D. Salber and G. D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," Anchor article of a special issue on context-aware computing in the Human-Computer Interaction (HCI) Journal, Volume 16 (2-4), 2001, pp. 97-166., 2001.
- 2) R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, PhD Thesis, University of California, Irvine, 2000.
- 3) L. Richardson and S. Ruby, "RESTful Web Services," O'Reilly Media, Inc, 2007.
- 4) OpenID, <http://openid.net/>
- 5) OAuth - An open protocol to allow secure API authorization in a simple and standard method from desktop and web applications., <http://oauth.net/>
- 6) 井上 武, 朝倉 浩志, 佐藤 浩史, 高橋 紀之, "REST アーキテクチャスタイルにおけるセッションについて," 信学技報, vol. 108, no. 458, IN2008-164, pp. 191-196, 2009 年 3 月.
- 7) J. Gregorio and B. de hOra, "The Atom Publishing Protocol," IETF, RFC 5023, Oct. 2007.
- 8) M. Nottingham and R. Sayre, "The Atom Syndication Format," IETF, RFC 4287, Dec. 2005.
- 9) FriendFeed, <http://friendfeed.com/>
- 10) OpenSocial, <http://code.google.com/apis/opensocial/>
- 11) Matthew Stabeller, Graeme Stevenson, Simon Dobson, and Paddy Nixon,

"Basadaeir: Harvesting User Profiles to Bootstrap Pervasive Applications," in adjunct proceedings of the Seventh International Conference on Pervasive Computing (Pervasive 2009), Nara, Japan, May 2009.