

## 携帯型 RDF レポジトリを活用した 最適コンテンツ提示に関する一検討

森川 大補<sup>†</sup> 菅谷 史昭<sup>†</sup>

† KDDI 研究所 〒356-8502 埼玉県ふじみ野市大原 2-1-15

E-mail: † {morikawa, fsugaya}@kddilabs.jp

**あらまし** ユーザ携帯電話を利用した活動履歴をもとにユーザプロフィールを生成・管理・活用する基盤の検討を行ってきた。ここでは、ユーザの能動的なアクションではなく、受動的なサービス享受を前提にしていた。このようなサービスの導入においては、ユーザの安心感が重要と考える。そこで、本稿では、ユーザプロフィール管理を携帯電話端末で行うことを提案し、それに基づくサービスシナリオ、携帯 RDF データ管理レポジトリの機能設計、および、携帯電話端末に実装するための基礎評価結果を示す。

### A Study of Content Presentation Suitable for User Profile Using RDF Repository Embedded on a Mobile Terminal

Daisuke MORIKAWA<sup>†</sup> Fumiaki SUGAYA<sup>†</sup>

† KDDI R&D Laboratories Inc. 2-1-15 Ohara Fujimino Saitama, 356-8502, Japan

E-mail: † {morikawa, fsugaya}@kddilabs.jp

**Abstract** A service provisioning framework, which generates, manages and utilizes user profile based on information gathered from a daily user of a user's mobile terminal, has been proposed and evaluated. It assumes that a user may have a "passive-type" service from the service provisioning system, not a "active-type" service which was invoked by a user's action. In the introduction of such context-aware service, it is important to brew up user's acceptance in the use of relevant user profile information. This paper proposes the user profile management in a mobile terminal and presents the service scenarios, the design of functionalities of mobile RDF data repository, and fundamental evaluation results in order to implement these functionalities on a mobile terminal.

#### 1. はじめに

携帯電話は、モバイルインターネット技術により、メールによるコミュニケーションや、Web アクセスによる情報取得が可能であるだけでなく、GPS による測位機能、非接触 ID カードによる決済機能、写真や映像を撮るカメラ機能なども具備するようになり、マルチメディア端末としての高機能化が進んでいる。そこで、我々は、このような機能を有する携帯電話を利用して、“その時々でのユーザの状況・活動”、そして、その集合体である“ユーザ活動の履歴”を取得・保持し、それを活用したサービス提供に関する研究を進めている[1]。

ユーザの携帯電話を利用した活動(履歴を含

む)をベースとしたサービス提供基盤は、図1に示す機能から構成される[2]。ユーザが所有する携帯電話によりユーザ状況やユーザの活動内容を取得する。取得した情報を、本研究では、ユーザプロフィールと呼び、プロフィールアグリゲータ(PA)と呼ぶミドルウェアが実装されたノードにおいて管理する。今までに検討を行ってきたシステムでは、プロフィール情報の管理主体と、サービスの提供主体を分離しており、サービス提供者が必要な情報をPAにアクセスして取得することを前提とする。これにより、サービス提供者は、自身で保持するユーザ情報に加え、PAから取得した情報を加味して、ユーザにサービスを提供することが可能

となる。

ここで、ユーザプロファイルはユーザ自身で管理するものとし、PA は個人のホームサーバの一機能としての実現を想定してきた。この際、プロファイルアグリゲータに求められる要求条件を以下に示す。

#### a) ユーザプロファイル更新機能

新たな情報が入ってきた際、管理しているユーザプロファイルを更新する機能。また、ユーザ履歴情報から特徴・傾向を抽出する機能なども含まれる。この機能は、どのような情報の管理形態であっても必要と考えられる。

#### b) トリガ通知機能

収集情報が目的の値と合致するタイミングを検出し、所望のサービス提供者へ通知する機能。なお、データの参照、更新のたびに条件をチェックするものとする。この機能は、サービス提供者とプロファイル情報の管理主体が異なること、かつ、サービスの起動がユーザの明示的なアクションによらないために必要となる機能である。したがって、ユーザの明示的なアクションによりサービスを起動させる場合は、本機能は必ずしも必要としない。

#### c) 検索機能

サービス提供者からの要求に対し、必要な情報を検索して返す機能。また、ユーザプロファイルがノード間をまたがり流通する際(サービス提供者への情報開示)のアクセス制御を並行して行う機能も有する。アクセス制御は、ユーザの明示的な確認作業を減らすために必要となる。

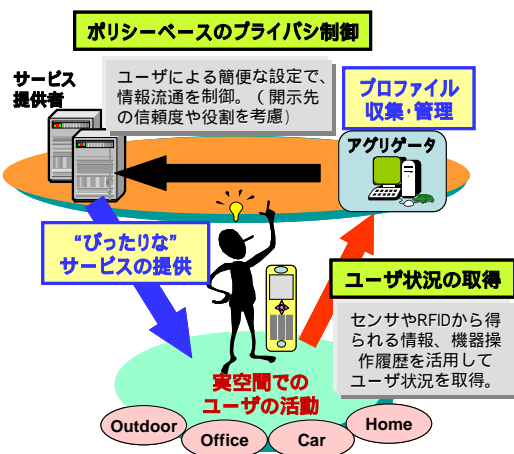


図 1 提案システムの概要

一方、このようなシステムの実用化に向けた検討を行う際、以下の点に注意する必要がある。

- ・ 携帯電話とプロファイルアグリゲータ機能を有するノードが異なると設定の手間が必要となる。
- ・ ユーザ情報が(ユーザの設定した)ポリシーにより開示されているとはいえ、このようなユーザ主導のサービス提供の初期段階においては、ユーザが常に情報を持ち歩き、サービスの提供タイミング(情報の開示タイミング)は、ユーザの能動的なアクションにより実施することで安心感が醸成される。
- ・ ユーザ情報を開示するタイミングで、開示内容をその都度確認できる機会を用意する。

本稿では、以上の点を考慮して、プロファイルアグリゲータ機能(のサブセット)を携帯電話端末に保持し、活用するためのシステムの検討を行う。本稿の以降の構成は、2章でサービスシナリオについて検討した結果を述べ、3章で携帯電話上での RDF データレポジトリを構成するための要件について述べ、4章で携帯電話上に実装するための基本評価結果を述べ、最後にまとめとする。

## 2. サービスシナリオ

現状の携帯電話の機能から取得できる(可能性のある)情報として、1) Web 閲覧履歴、メール送受信履歴、アプリ利用履歴、2) FeliCa などを利用した決済履歴、3) GPS による位置履歴、4) Bluetooth や IrDA などの近接通信を利用した接続履歴、などが想定される。また、将来は、ホーム環境にある STB (Set Top Box) などをはじめとする情報家電の操作履歴、外部機器と連携した機器利用履歴、周辺をセンシングするセンサ機器とのアドホックな連携による情報取得が想定される。したがって、このような情報の適切な管理と活用法の検討が必要となる。

本稿では、ネットワーク上、あるいは相手の端末上にある情報(コンテンツ)を取得する場合に限定したサービスシナリオを対象とし、2つのサービスシナリオを検討する。1つ目は、個人のプロファイル情報を外部に開示することなく、サービスが完結する方法である。2つ目は、個人のプロファイルを開示し、複数のユーザの結果から適切な情報を取捨選択する方法である。さらに、友人同士で取得したコンテ

ンツを互いに検索して閲覧しあう方法も 1 つの利用形態として述べる。

## 2.1 シナリオ 1：プロフィール情報を非開示

ユーザのプロファイル情報を外部に出さない場合は、ユーザにとって最も安心なケースである。この場合、取得した情報（コンテンツ）を端末内でフィルタリングもしくは並べ替えることにより、ユーザのプロファイルにあった情報を提示することが可能になる。この方式の実装例を図 2 に示す。

この例では、サービスの起動に FeliCa インタフェースを利用している。まず、街角にある FeliCa リーダ・ライタにユーザが携帯電話端末をかざすことによりサービスを起動する（2-1）。続いて、FeliCa リーダ・ライタから接続先情報配信装置のネットワークアドレス情報などを取得して接続する。その結果、RSS 拡張形式を利用したお勧め情報リストを取得する（2-2）。携帯内で管理するプロフィールに応じて、取得情報がフィルタリングされて表示される（2-3）。なお、ここで、配信する情報は、携帯内部で保持するプロフィール情報との比較を容易とするために、RSS 拡張形式を想定する。

また、FeliCa を利用した情報配信は、携帯電話をかざすリーダー・ライタ毎に識別できることから、場所に応じた情報配信を行うことが可能である。例えば、テックファーム社の TOWN POCKET[3]や、NTT ドコモのトルカ[4]などがあるが、場所依存ではあるが、配信情報はユーザ毎にカスタマイズされてはいない。



図 2 シナリオ 1：プロフィール情報非開示

## 2.2 シナリオ 2：プロフィール情報を開示

ユーザのプロファイル情報を開示する場合は、プロフィールを活用した情報配信方法がいく

つか考えられる。その 1 つとして、複数ユーザのプロファイル情報をもとに適切な情報をサーバ側で選択して配信する方式を検討する。この方式の実装例を図 3 に示す。

この例では、ユーザは気に入った情報をユーザプロフィールとともに、ソーシャルブックマークサーバ (SBM サーバ) に登録する (3-1)。SBM サーバに参加するユーザが増えれば増えるほど、ユーザプロフィールに適合する情報が確度高く決定できると考えられる。ユーザが情報を取得する際には、プロフィールを送信し、プロフィールに合致するお勧め情報リストを取得する (3-2)。

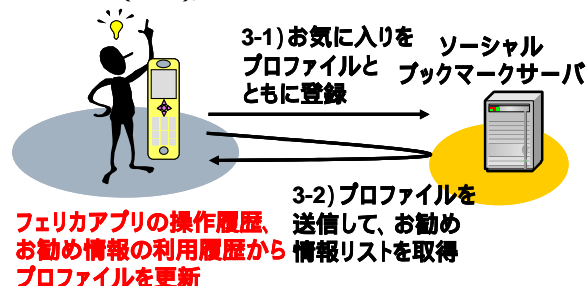


図 3 シナリオ 2：プロフィール情報開示

## 2.3 シナリオ 3：携帯電話端末同士の検索クエリによる情報交換

上記シナリオで端末に保持した情報（コンテンツ）を互いに交換する状況を想定する。その際、すべてのコンテンツを取得するのは得策ではないので、相手が保持する情報のうち、自身のプロフィールにあったものや、設定したパラメータ値に合致する情報を取得するために、検索要求を行い、条件に合致した情報のみを取得する。

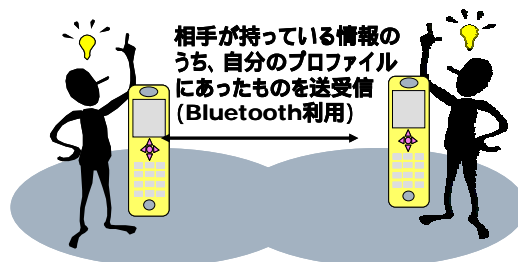


図 4 シナリオ 3：検索クエリによる情報交換

以上、携帯電話端末にユーザプロフィールの管理ができるようになった場合のサービスシナリオをいくつか示した。次章では、携帯電話上に RDF データを管理・利用するための要件について述べる。

### 3. 携帯 RDF データ管理レポジトリ

本章では、提案システムの主な構成要素について述べる。

#### 3.1 RDF ( Resource Description Framework ) を用いた取得情報の管理

提案システムでは、RDF ( Resource Description Framework ) [5]によるデータモデリングを行う。ここで、RDF を利用するのは、1 件のデータを複数のフィールドの集合で表現する RDB (Relation Database) では、同一形式の情報を多数処理することには向くが、種別が多くなるとその取り扱いは難しいとされていること、また、スキーマの変更が頻繁に発生する事例に対しては、XML 文書そのままツリー構造で格納可能な XML-DB の利用が有効といわれていることからである。

最近では、RDF を用いた語彙して、ホームページの要約情報を配信するための RSS (RDF/Rich Site Summary) [6]、Web 上で自身のプロフィールや友人に関する情報を公開するための FOAF[7]、書籍情報のメタデータである Dublin Core[8]などが広まっているが、本研究では、標準化されていない語彙については、独自に定義した。また、Semantic Web で用いられている RDF、RDF スキーマ、オントロジ記述言語の OWL ( Web Ontology Language ) [9]を用いて、リソース同士の関連性や、クラスの継承関係の記述している。

RDF は、ある URI で示されたリソースに対する記述を規定するもので、( s, p, o ) (ここで、s : Subject, p : Property, o : Object ) の三組により記述できる。もとは、Web 上のリンクなどのリソースを対象にした記述言語であるが、これを、実世界のモノや人間、あるいは、人間の活動、といったあらゆるものを、システム内のリソースとして表現することが可能である。以上から、RDF を用いて、情報種類の多様性に対応するデータモデルを作成した。

ここで、コンテンツに付随する拡張 RSS の形式の例 (一部) を図 5 に示す。Subject は Resource で表現されるため丸印で示されており、Object は、階層的に連結される場合は Resource をとり、連結されない場合は Literal 値をとるため、前者は丸印、後者は四角で標記している。また、Property は、矢印の上に標記している。なお、ここでは、定義した Property の詳細については割愛する。

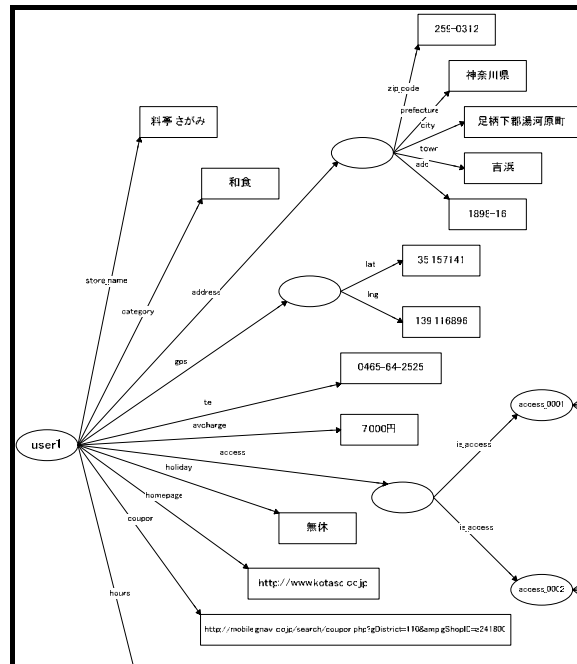


図 5 コンテンツに付随させる拡張 RSS 形式

#### 3.2 RDF 操作機能

我々は、Jena2[10]をベースに携帯への実装を検討した。なお、オープンソースで提供されている Jena は、SemanticWeb アプリケーションを作成するための Java フレームワークであり、RDF を読み書きするための API、RDF モデルの入出力等が提供されている。まず、RDF 操作への要求機能は以下の通りである。

- ・ 履歴データ管理
- ・ クエリ処理

なお、携帯電話上で RSS リーダが実現されているが、これは RSS ファイルを閲覧する機能が主であり、より汎用的な RDF/XML 形式のデータを生成・保持する機能、目的の RDF/XML 形式の情報を検索する機能、データ要素どうしを比較するなどの演算機能を設計・実装している点が、従来のソフトウェアと異なる点である。

Jena で提供されているメソッドのうち、サポートする関数を表 1 に示す。これらの機能を満たすことによって、基本的な RDF データの処理はプログラム上で行うことが可能である。クエリ処理のための記述については、RDF では、SQL に近い RDQL[11]や SPARQL[12]が標準化されているが、携帯電話端末に利用するアプリケーションサイズの制約から、頻りに利用するクエリ種別のみを実装することとした。汎用的なクエリ言語の設計・実装は今後の課題である。

表1 Jena2 との対応状況

getProperty (Resource s, Property p)	(s,p,o)の組合せでsがs、pがpとなるstatementを返す
getProperty(nameSpace, localName)	与えられたURIを有するpropertyのinstanceを返す
getResource(uri)	与えられたURIを有するresourceのinstanceを返す
listObject()	RDFモデルの中のすべてのobjectのリストを返す
listObjectsOf Property (Property p)	与えられたpropertyに関するすべてのobjectを返す
listObjectsOf Property (Resource s, Property p)	そのresource, propertyに対応するすべてのobjectを返す
listStatements()	すべてのstatementを返す
listStatements (Resource s, Property p, RDFNode o)	指定したパターンのstatementを探す
listStatements(Selector s)	セレクタにマッチするstatementを探す
listSubjects()	Statementに関わるsubjectのすべてを返す
listSubjectsWithProperty (Property p)	与えられたpropertyに関するすべてのsubjectを返す
listSubjectsWithProperty (Property p, RDFNode o)	与えられたpropertyと値を保持するすべてのobjectを返す
query(Selector s)	クエリにマッチする(s,p,o)の組合せを含むRDFモデルを生成

#### 4. 基本性能評価結果

KDDIの提供するW32S型の携帯電話移動機を利用して、携帯型RDFレポジトリの実装に必要な基本性能を評価した。

計測方法は、RDF/XMLデータをパースし、初期化したBREWデータベースに対し登録する処理を100件繰り返す。ここで、**書込み処理時間**を、パース開始から100件のレコードの書込みが完了するまでの時間と定義する。また、検索開始からRDFファイルの出力が完了するまでの時間を**読み込み処理時間**と定義する。ここで、読み込み方法の種類によって、表2に示す4つのパターンを計測した。計測結果を図6に示す。計測結果より、書込み時間については、既に登録されているレコード件数とは関係がな

く約1000msec程度と一定している。一方、読み込み時間は、検索条件に合致するデータの保持箇所に比例して、検索時間がかかることがわかる。また、レコード数が1700件でも2000msec以内に収まっている。なお、書込みでは、基本的なRDF/XMLの書式で、2つのstatementが記述されているものを取上げた。

続いて、RDF/XMLファイルの中身を10ノード単位で追加し、パースの関数処理が完了するまでの時間を計測した。追加するノードについては、名前空間、ノード名は重複し、リテラルが重複しないよう設定した場合(パターン)と、名前空間、ノード名、リテラルすべてが重複しないように設定した場合(パターン)を計測した。測定結果を図7に示す。一般的に、パース処理時間は、RDFファイルに含まれるノード数に比例して長くなる。また、ノード数が360のときでもおおよそ300msec以内の処理である。以上のデータは、携帯型RDFのデータ保持数を検討する際には、ユーザの待ち時間を考慮する際の指標となる。

表2 計測条件

パターン	登録済データベースから必ず先頭レコードにヒットする検索条件を設定し、検索結果をRDFファイルに出力
パターン	登録済データベースからヒットしない検索条件を設定し、検索結果をRDFファイルに出力
パターン	登録済データベースから必ず最終レコードにヒットする検索条件を設定し、検索結果をRDFファイルに出力
パターン	登録済データベースから必ず最終レコードにヒットする検索条件を設定し、かつ、データベースで提供されているインデックスを利用した検索結果をRDFファイルに出力

#### 5. まとめ

我々は、ユーザ状況や活動に適合するサービス提供を目的として、ユーザプロフィール管理活用基盤を提案・検討してした。そこでは、ユーザプロフィールの素となる情報を主に携帯電話端末から収集しているが、その収集ノードとプロフィール情報の管理ノードが異なるため、初期設定の手間、ユーザの能動的なアクションによるものでなく受動的なサービス享受を想定していたため、ユーザ主導のサービス導入時点では、ユーザの安心感の醸成が重要となる。

そこで、ユーザプロファイルの管理を携帯電話端末で行うことを前提とし、それに基づいていくつかのサービスシナリオの検討、携帯RDFデータ管理レポジトリの機能設計を行った。さらに、携帯電話端末での処理性能について基礎データを取得し、実際に携帯電話に実装可能な目処を立てた。今後は、サービスシナリオを含む詳細設計と実装を進め、評価を進める予定である。

謝辞 日頃よりご指導いただき KDDI 研究所、秋葉所長、中島執行役員に感謝致します。

### 文献

- [1] 森川他, “複数の情報源から集約したユーザプロファイルの管理活用基盤の検討,” 信学技報, NS2003-321, pp.143-148, 2004年3月.
- [2] Morikawa et al., “A proposal of user profile management framework for context-aware

service,” In Proc of SAINT2005 Workshop, Jan. 2005.

- [3] TOWN POCKET, <http://www.townpocket.jp/view/pc/about01.html>
- [4] トルカ, <http://www.nttdocomo.co.jp/service/imode/osaifu/toruca/>
- [5] RDF, <http://www.w3.org/RDF/>
- [6] RSS, <http://web.resource.org/rss/>
- [7] FOAF, <http://www.foaf-project.org/>
- [8] Dublin Core, <http://dublincore.org/>
- [9] OWL, <http://www.w3.org/TR/owl-features/>
- [10] Jena, <http://jena.sourceforge.net/>
- [11] RDQL, <http://www.w3.org/Submission/RDQL/>
- [12] SPARQL, <http://www.w3.org/TR/rdf-sparql-query/>

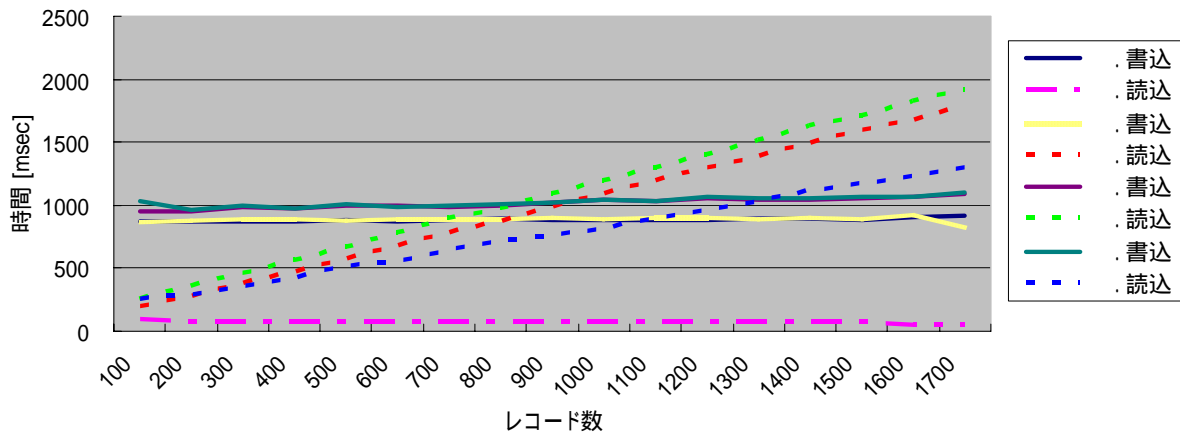


図6 RDF/XMLの読み込みおよび書き込み処理性能(レコード数に対する処理時間)

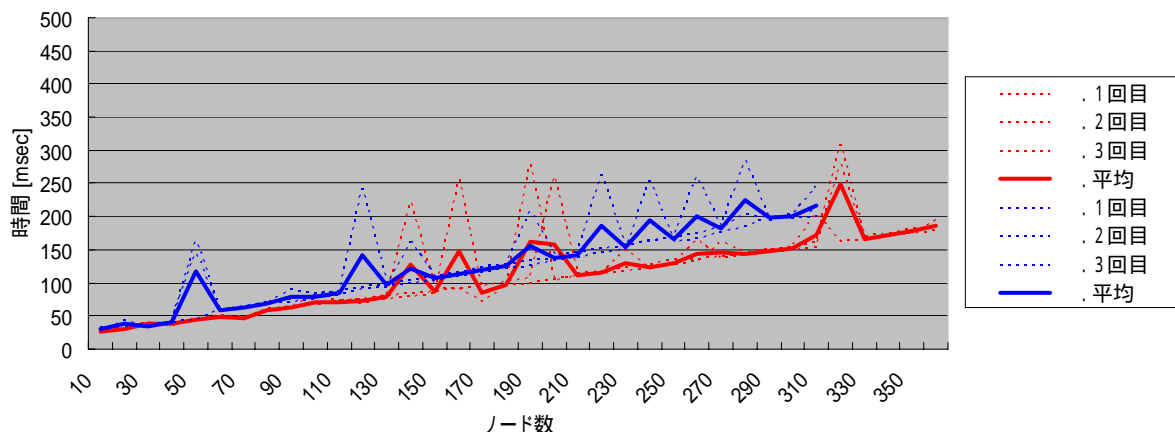


図7 RDF/XMLファイルのノード数に対するパース処理時間