

# RDF 記述化された個人環境プロフィールの統合利用について

本庄勝 森川大補 山口明 大橋正良

KDDI 株式会社

E-mail : {ma-honjou, da-morikawa, ma-oohashi, ai-yamaguchi}@kddi.com

## 要旨

著者らは、ユーザ個人に関する情報、ユーザ周辺の環境情報などをユーザの状況に応じて収集し、利活用するためのサービスプラットフォームについて検討を進めてきた。本稿では、RDF (Resource Description Framework) で記述されたそれらの取得情報に対し、固有の URI (Uniform Resource Identifier) を割り当てていくことで情報の統合化を図るための手法を検討したので報告する。我々の取り扱う情報 (個人の契約情報や位置情報、物品情報、電子決済情報、環境情報など) をここでは個人環境プロフィールと呼ぶ。個人環境プロフィールはメタデータで記述されており、それぞれがユーザの生活に関わる情報である。これらの情報の中には互いに重複した情報 (リソース) も含まれている。そこで同一のリソースと判断されるものには同じ URI を割り当てることにより、ユーザにとって統一的な情報管理手法を提案する。

## A Study of Integration of RDF Described Personal Environmental Profiles

Masaru HONJO Daisuke MORIKAWA Akira YAMAGUCHI Masayoshi OHASHI

KDDI Corporation

E-mail : {ma-honjou, da-morikawa, ma-oohashi, ai-yamaguchi}@kddi.com

## Abstract

In this paper, we discuss the integration method for individual RDF (Resource Description Framework) described Personal Environmental Profiles. Here we call meta-data information relative to user daily life (personal contract information, location, artifact, electrical receipt, sensing information and so on) as Personal Environmental Profiles. We consider that there are same information on these profiles. So, we propose URI (Uniform Resource Identifier) integration method so that user integrates and manages these profiles. By giving the same URI to the resources which have same properties and literals, we can realize this information integration.

## 1 はじめに

将来のユビキタス社会の一つの実現例として、ユーザ個人に関わる情報やユーザの生活する周辺地域の環境に関する情報などがメタデータで記述化され、ユーザの状況に応じたコンテキストウェアサービスが提供される世界が考えられている。

現在最もシンプルな形態は、GPS 情報による

位置依存型サービスだろう。携帯端末で取得した GPS 情報はサービスプロバイダに渡され、ユーザは対応した地図や近隣のショップ情報を得ることができる。また将来携帯電話に RFID tag リーダが実装されれば、ユーザが手にした商品の ID を読取ることで、ユーザが興味を示している商品の情報に紐付けされた情報をネット上で取得して表示することもできる。このようにメタデータの生成

は今後、次々に増加していくだろう。さらに収集した情報を個人で管理しておけば、自分の興味を示していた情報を個人サーバ内で検索し、後でその一覧と商品の価格リストの表示を見するというようなシチュエーションも生まれてくる。

モノの属性情報、ユーザの各種特徴をデータ化した情報等をプロフィールと呼ぶことにするが、著者らは、こうしたプロフィール情報を個人レベルで統合的に管理しそして活用するための個人環境プロフィール統合プラットフォームの検討を進めてきた [1]。ここでは特にユーザの個人情報や契約情報、位置情報、物品情報、環境情報、また電子決済情報やユーザの使用した携帯端末やホーム PC の利用履歴といったログ情報などを集約、体系化したものを総称して個人環境プロフィールと呼んでいる。

個人環境プロフィールの元となるプロフィール情報は各種のサービス事業者やユーザの所属する組織若しくはユーザ自身によって提供されると考えられる。それぞれを見比べたとき、お互いに情報が重なり合っていることがあると考えられる。例えば、ユーザが手にしたときに取得した商品トレース情報と実際に購入したときに取得する電子決済情報（電子レシート）とは同じ商品に関する情報が含まれる。

個人環境プロフィールの管理において、様々なプロフィール情報を単に集約するだけであると、情報量が膨大になって、有益な情報が埋もれてしまって有効に活用できない恐れがある。例えば、購入した商品について、過去に取得した関連情報を検索したい場合もある。それゆえ個人環境プロフィールの構築では各要素間に含まれる情報を統合して管理し、同一物であると体系化して整理することで、新たな価値を生成することを考慮している。

そこで本稿では、これまでに検討してきた [1] の中でも特に収集した個人環境プロフィール情報の統合手法についてに焦点を絞り、検討を深める。収集したプロフィール情報は個人環境プロフィールに変換する際に RDF (Resource Description Framework) [7] によって記述される。そこで商品などといった対象物を RDF のリソースとして扱うことで、その URI (Uniform Resource Identifier) を割り振ることを考える。RDF では同じ情報 (リソース) に対して同じ URI を割り当てることでお互いの情報を紐つけることができる。RDF で記述された個人環境プロフィールでは、リソースの属性を表すプロパティを比較すると同一のものに対しては同じリテラル値を持つと考えられる。そこで各リソースに対し持ちうるプロパティの整合を取ることによってリソースの統合を図ることにする。

## 2 プロファイル情報

ユーザの生活に関わる情報、例えばユーザの個人プロフィールやサービスプロバイダとの契約情報、電子決済情報や物品の詳細情報、各エリアで提供されるサービスメニュー情報といったメタデータ情報を我々はプロフィール情報と呼んでいる (図 1)。

各種プロフィール情報は、ユーザの生活の中で逐次取得されるので、見方を変えれば、これらのプロフィール情報は人の生活における一つのログとみることもできる。一つの利用形態としては、収

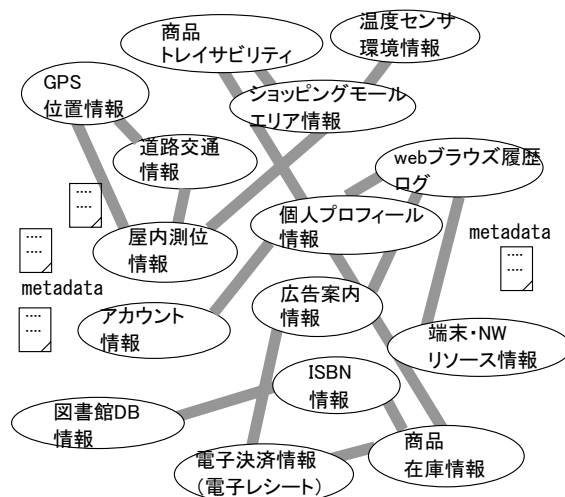


図 1: 各種プロフィールとお互いのつながり。

集したプロフィール情報を蓄積し分析することで、ユーザの特徴を抽出技術に適用することもできるだろう。

プロフィール情報は、互いに異なる情報提供者から得られる。それゆえ情報源そのものは独立である。しかし収集したプロフィールから特徴抽出できることから分かるように、収集した情報の中には互いに類似性を有している。はじめにでも述べたように、ユーザの手にした商品情報のメタデータと電子レシートのメタデータには同じ商品の情報が含まれていると考えられる。そこで関係した情報を図 1 のようにお互いに関連付けすることができれば、ユーザにとってより利便性が高まり、自分のログから関連情報を引き出したり、また知識データベースのような情報を生成することが可能になる。

個別の情報源から収集されるため、これらのメタデータは収集した時点では、それぞれの関連付けはなされていない。そこで個人レベルで情報を収集し、個人環境プロフィールとして情報の統合管理することを検討している。

## 3 個人環境プロフィール

我々がこれまでに検討してきたシステムの構成を図 2 に示す [1]。ここでプロフィール情報の提供者をプロフィールコレクタ、また個人がプロフィール情報を集約し、個人環境プロフィールを構築する側をプロフィールアグリゲータと呼ぶ。

プロフィールアグリゲータでは、個人環境プロフィールを RDF 形式で管理する。取得された各プロフィール情報に対し、特徴を抽出し体系化したものを一つの個人環境プロフィールとして整形し、またユーザの行動履歴や、手にした商品の情報といった各種のログ情報を個人環境プロフィール DB に蓄積する。プロフィール情報は入力時に RDF に変換される。そこで情報を統合化するために、リソース統合装置を設けて、個人環境プロフィールの要素間の関連付けを行う。

なお、プロフィール情報の流通については、RDF

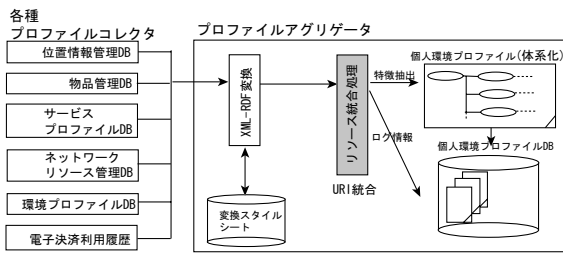


図 2: 個人環境プロフィール構築のシステム構成。

をベースとして記述したものが今後広まるものと考えている。コンテンツ流通においても CC/PP[3] や RSS[4] などといった RDF 形式をベースとした標準化が進められている。ただ、プロフィールコレクタからの情報に関しては、XML ではあっても RDF ではないプロフィール情報も考えられるため、この点についてはこれまでの検討 [2] により、スタイルシートを用いることで、XML 形式を RDF 形式に変換することを考えている。

### 3.1 RDF 記述化された個人環境プロフィール

個人環境プロフィールを RDF モデルで表した例を図 3 に示す。図 3 (a)、(b) はそれぞれ我々が扱う電子タグによって収集された商品の物品情報と、商品を購入した際に得られた電子レシートの例を示している。RDF の記述では、リソース、プロパティ、リテラルをトリプル（基本要素）として表現する。図 3 (a) の例では、ユーザががざした商品の名前 (kprof:name) や管理コード (kprof:epc)、製造国 (kprof:madein)、製造日 (kprof:dateofmanufacture) といった情報がプロパティとして記述されており、商品そのものをリソースとして扱っている。図で rdf:type はそのリソースのクラスが何であるかを意味するプロパティである。この場合、リソースは商品 (kprof:goods) であることを記述している。図 3 (b) も同様である。ただ電子レシートの場合ではレシートそのもの、購入した商品、購入した場所の三種類のリソースとそれぞれのプロパティ情報が記述されている。

RDF では URI によって各リソースの識別子を与える。ただし、図 2 で RDF に変換した時点ではこの URI は指定されていない。この場合は図 3 のように各リソースを表す楕円の内部は空白で表現される。

#### 3.1.1 個人環境プロフィールの比較

図の (a) と (b) を見比べると、斜線の部分で同じ情報を表している。実はこれらは同一のリソースを示している。そこでそれらを同一リソースと判断し、紐づけてやれば、物品情報と電子レシートを関連付けることができる。これにより電子レシートの購入価格情報から物品情報を取り出し製造国や製造日を引くことができる。

我々はリソースにユーザ独自の URI を割り振ることでその実現を図る。商品に同じ URI を与え、(a)、(b) をパーサで同時に処理すれば、互いに情報が結合した形で RDF モデルを表現するこ

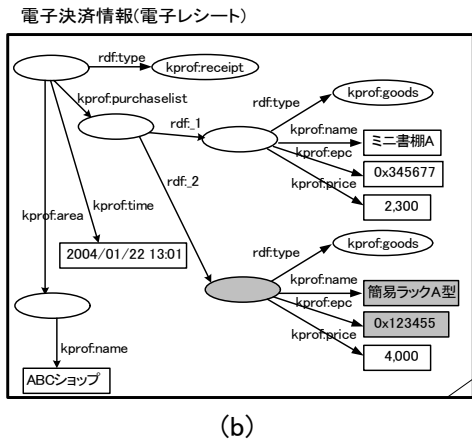
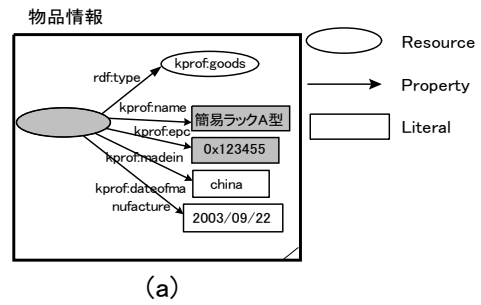


図 3: 個人環境プロフィールの RDF モデル。

表 1: 個人環境プロフィールと含まれるリソース。

個人環境プロフィール	含まれるリソースの例
個人プロフィール	人
契約情報	契約書, 人, サービス
電子決済情報	決済表, エリア, 商品
物品情報	商品
エリア情報	エリア, サービス
...	...

とができる。図 4 はその統合例である。商品には kprof:goods\_1 などといった URI を割り当てている。このようにして同一の情報をお互いに紐づけて情報の統合を図る。

商品以外にも個人環境プロフィールの中には、幾つか情報が統合できるものが含まれている。これまでの検討の中で個人環境プロフィールの種類と含まれると考えられるリソースを表 1 に示す [1]。この例の中では人や商品、エリア、サービスなどが統合できるリソースと見ることができる。

#### 3.1.2 個人環境プロフィールのインスタンス

図 3 (b) に対応した RDF のインスタンスを図 5 に示す。RDF 変換後の各リソースの記述は、次のような書き出しで始まっている。

```
< rdf:description >
```

RDF でリソースに URI を指定した場合は、例

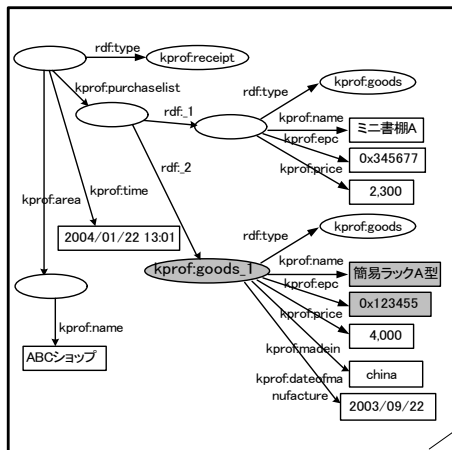


図 4: 統合された個人環境プロフィール.

例えば次のように記述される.

```
< rdf:description about = "kprof:goods_1" >
```

つまりこの部分に URI 情報を追加することで、リソースに URI を与えることができる。本稿の提案では、具体的にはこの部分に、複数のプロフィール間で共通の URI を書き加えることになる。

本手法では、リソースのプロパティの比較を行って両者が同一リソースと判断されると統一の URI を書き加える。異なると判断された場合には、異なる URI を割り当てる。本手法では RDF の形式を変えてしまうことはせず、description のタグ部分のみを書き換える。この URI 統合の更新手順については次節にて説明する。

なお図 5 における各種プロパティやクラスの定義 (語彙) については、本開発上でスキーマ定義をしている。kprof というのは我々が定義した語彙であることを表す接頭辞である。商品に関するスキーマの例を図 6 に示す。この例では 8 種のプロパティが定義されており、rdfs:domain は基本プロパティの一つで、あるプロパティがどんなクラスのリソースに適用できるかを表している。また rdfs:range はそのプロパティがどんなクラスの値をとるかを示しており、この場合全てのプロパティがリテラル値をもつことを示している。

ただ独自の語彙であると汎用性がなくなるので、標準的に利用されている Dublin Core[5] や vCard[6] のような語彙を個人環境プロフィールに取り入れることを検討している。URI の統合についても特に共通の語彙で表現されている方が統合の可能性が高いと言える。

## 4 URI 統合アプローチ

収集した個人環境プロフィール情報を蓄積し、一元的に管理するための URI 統合方式を述べる。[1] では、体系的な個人環境プロフィール部分を提示しているが、ユーザの手にした商品の情報、過去の履歴等はログ情報としてデータベースに蓄積される (図 2)。本稿では本開発における個人環境プロフィール DB に蓄積する際の URI 統合手法を示す。

```
<?xml version=" 1.0" encoding=" Shift_JIS" ?>
<!DOCTYPE rdf:RDF[
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY kprof "http://profile.weng.kddlabs.co.jp/ns/kprof#" >
]>
<rdf:RDF xmlns:rdf=" &rdf;" xmlns:rdfs=" &rdfs;" xmlns:kprof=" &kprof;" >

<rdf:Description>
<rdf:type rdf:resource=" kprof:receipt" />
<kprof:time>2004/01/22 13.01</kprof:time>
<kprof:area>
<rdf:Description>
<kprof:name>ABCショップ</kprof:name>
</rdf:Description>
</kprof:area>
<kprof:purchaselist>
<rdf:Bag>
<rdf:li>
<rdf:Description>
<rdf:type rdf:resource=" kprof:goods" />
<kprof:name>ミニ書棚A</kprof:name>
<kprof:epc>0x345677</kprof:epc>
<kprof:price>2,300</kprof:price>
</rdf:Description>
</rdf:li>
<rdf:li>
<rdf:Description>
<rdf:type rdf:resource=" kprof:goods" />
<kprof:name>簡易ラックA型</kprof:name>
<kprof:epc>0x123455</kprof:epc>
<kprof:price>4,000</kprof:price>
</rdf:Description>
</rdf:li>
</rdf:Bag>
</kprof:purchaselist>
</rdf:Description>
</rdf:RDF>
```

図 5: 個人環境プロフィールのインスタンス.

これまでに検討してきた URI 統合の基本的な流れを図 7 に示す。ここでユーザが取得した個人環境プロフィールを入力 RDF プロファイル、既に DB に蓄積されているものを蓄積 RDF プロファイルと呼ぶことにする。個人環境プロフィール DB には、各種のログが蓄積されており、蓄積 RDF プロファイルには既に URI が割り与えられている。リソース統合処理では、入力 RDF プロファイルを個人環境プロフィール DB に蓄積する前に既存のリソースと同一のものがないかどうかの判断を行い、URI を割り当てる機能を有する。

本検討では、URI はユーザ固有の接頭辞にユーザが管理するリソースの ID を割り与えるものとする。

なお、実際の処理は、各 RDF プロファイルは XML ベースで記述されているものとし、DB への検索においては RDQL 等のクエリ言語等を用いて行うことを想定している。

入力 RDF プロファイルに URI を付与し、個人環境プロフィール DB に蓄積するまでの全体的な流れは以下のとおりである。

- Step. 1 入力 RDF プロファイルに含まれる全てのリソースに対して rdf:type 属性のクラスを抽出する。例えば図 7 の場合、入力 RDF プロファイルに含まれる Class:I, Class:II が抽出される。
- Step. 2 次に蓄積 RDF プロファイルから同一のクラスであるリソース及び、そのリソースの持つプロパティを検索する。例えば、クラスが Class:I である URI を検索するには、次のような RDQL を用いる。

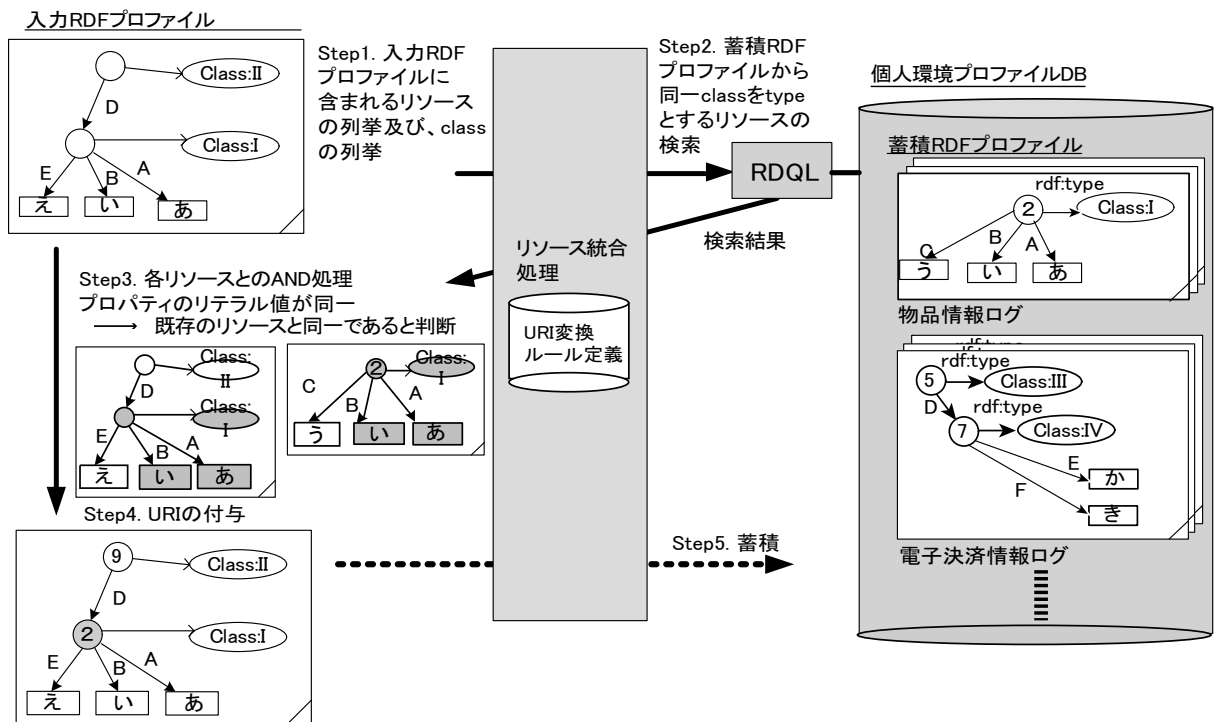


図 7: URI 統合及び蓄積の流れ.

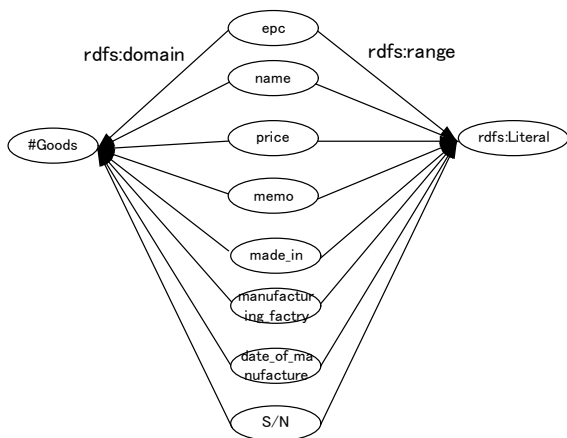


図 6: 商品 (#goods) に関するスキーマ定義.

```
Select ?x from < http://... >
where (?x, < rdf:type >, ?type)
and ?type = < http://...#Class:I >
```

ここで URI が #2 のリソースが検索できたものとし、次にプロパティを抽出するためには次のような RDQL を用いる.

```
Select ?y ?z from < http://... >
where (?x, ?y, ?z)
and ?x = < http://...#2 >
```

Step. 3 次に入力 RDF プロフィールと検索して得られた蓄積 RDF プロフィールとのパターンマッチングを行う。各クラス毎にパターンマッチングを適用するために、URI 変換のためのルール定義を用意した。同一の情報源から得られた RDF プロフィールであれば、形式は同一であるため、ルール定義で記述することで、入力 RDF プロフィールの適切な description 記述に URI を書き込む。またルール定義には比較するプロパティのリストが記述されており、比較する中でも同プロパティに対してリテラル値が同じであった場合、同一のリソースであると判断する。図では A, B のプロパティに対し、“あ”、“い” のリテラルが対応しているため同じ URI の #2 が割り当てられることになる。

Step. 4 この時点で入力 RDF プロフィール内のリソースに URI を割り当てる。今までに存在していなかったものと判断された場合には、ユニークな URI を新規に割り当てて全てのリソースの URI 割り当てを終了する。

Step. 5 最後に入力 RDF プロフィールを個人環境プロフィール DB に蓄積する。入力 RDF は一つの行動ログとして個人環境プロフィール DB にカテゴリに分割されて保存される。

URI 統合の中で、何をもちて同一とみなすのかは一つの課題である。本検討内容では、同一クラスのリソースの中で共通のプロパティとリテラル値を一つでも持てば、統合の対象としているが、誤って統合してしまうことも考えられる。そこでどの

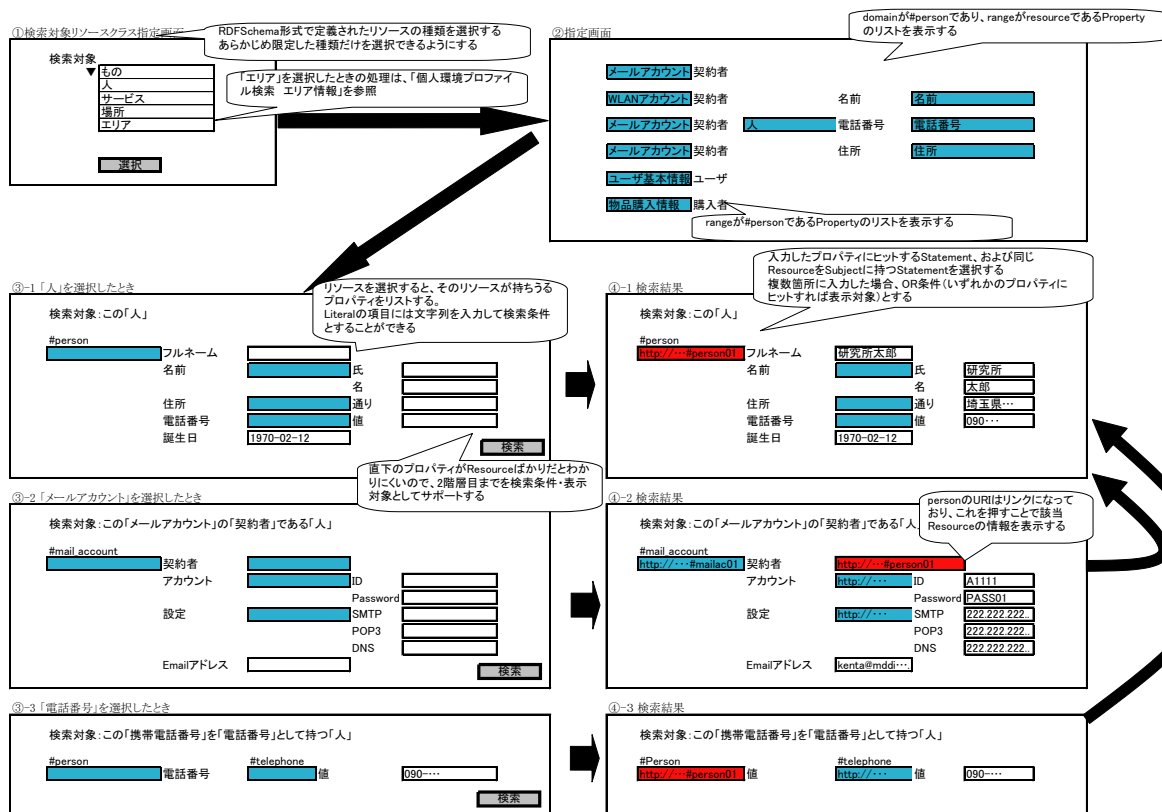


図 8: 個人環境プロフィール検索アプリケーション試作画面。

程度同一性を有しているかの指標（類似性）を与えることが必要である。

## 5 検索アプリケーション

今回、URI 統合の動作を確認するために簡単な検索アプリケーションを試作することにした。概観を図 8 に示す。

ユーザの検索対象を選択し、それに関する情報が各種個人環境プロフィール間でお互いに関連付けられているかを検証することを目的とする。本報告では、試作モデルの概要までを説明することにする。

まず画面①はこの検索アプリケーションのトップ画面である。画面①においてユーザは自分の個人環境プロフィールの中から調べたいものを選択する。ここでは RDF のスキーマで既に定義されているクラスの中でユーザの検索対象（もの、人、サービス、場所、エリア）を表示させることにした。ここで“人”を選択した場合、画面②の検索指定画面が表示される。画面②には、本開発上で定義しているスキーマの中で、“人”のクラスである #person を domain（主語）もしくは range（目的語）としている全てのプロパティのリストを表示している。画面左の“契約者”、“ユーザ”、“購入者”が、#person を range とするプロパティ、“名前”、“電話番号”、“住所”が #person を domain とするプロパティである（これが商品の場合、図 6 のスキーマに応じたプロパティのリストが表示される。

）。これらはスキーマによって既に定義されているものとする。

画面②で“人”を選択した場合、画面③-1 に遷移する。この画面でユーザは調べたい人について知ろうる情報を空白の欄に入力する。そして検索ボタンを押すと、画面④-1 を表示し、入力した情報にマッチした“人”の検索結果が表示される。検索結果では空白には、検索した“人”の情報が補填される。内部的には、入力したプロパティのリテラル値にヒットするステートメント、及び同じ URI を Subject に持つステートメントを個人環境プロフィール DB から検索している。画面③-1 で複数箇所に入力した場合は、OR 条件（いずれかのプロパティにヒットしたものを表示）する。検索で複数ヒットした場合には画面④-1 を複数分表示する。なお、画面③で検索対象の持つリテラルが全くなく、ユーザが入力できる欄がない場合も考えられる。そのため、リテラルの入力欄は 2 階層まで表示させるようにする。

画面②で“メールアドレス”を選択した場合には、メールアドレスを domain とするプロパティを表示する（画面③-2）。同様にユーザが知っている情報を入力することで、その情報を持つリソースを検索することができる（画面④-2）。画面②で“電話番号”を選んだ時も同様に画面③-3 に遷移して電話番号を入力することで電話番号から検索した結果を画面④-3 のように表示する。

画面④-2、画面④-3 の検索結果の中で、リソースの欄には、ユーザが調べたかった“人”の URI が

表示されている。ここでその URI に紐付けられた情報をさらに調べるためにリンクを張る。ユーザはそのリンクをクリックすることで、さらにその“人”の情報を検索することを可能とすることにした(画面④-1 に遷移)。

## 6 まとめ

本稿では、個人環境プロファイルの統合プラットフォーム [1] の中で、特に RDF 記述化された個人環境プロファイルの URI 統合手法について述べた。同じリソースを表現するものであれば、同一のプロパティとリテラル値を持つものを含んでいると考えられる。そこで各リソースに対してプロパティの整合をとることで、同一と判断されるものには、個人環境プロファイルに記述されているリソースに URI の記述を加えることで情報の統合を図った。

今後の課題としては、URI 統合の際に判断基準となる整合の精度や類似度を数値化することが考えられる。

### 謝辞

日頃よりご指導いただき KDDI 村上本部長、濱井部長に深く感謝致します。なお本研究は、総務省からの委託研究の成果である。

## 参考文献

- [1] 森川 大補, 本庄 勝, 山口 明, 大橋 正良, “ユーザ状況に基づいたプロファイル体系化およびその活用に関する一検討,” 情処学 UBI 研報, No. 2, pp. 219–224, Nov. 2003.
- [2] 本庄 勝, 森川 大補, 山口 明, 大橋 正良, “プロファイル情報とその統合利用技術に関する一検討,” 2003 信学ソ大, No. B-15-20, Sept. 2003.
- [3] F. Reynolds, J. Jjelm, S. Dawkins, S. Singhal, “Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation”, W3C Note, 27 July 1999. <http://www.w3.org/TR/NOTE-CCPP>
- [4] Dan Brickley, et al., “RDF Site Summary (RSS)1.0,” RSS-DEV Working Group, 2000. <http://purl.org/rss/1.0/spec/>
- [5] <http://dublincore.org/documents/dces/>
- [6] <http://www.w3.org/TR/vcard-rdf>
- [7] <http://www.w3.org/RDF/>
- [8] <http://www.ietf.org/rfc/rfc2396.txt>