

ユーザオントロジーに基づく適応的リソース選択のためのフレームワークの実現

Construction of Framework for Personalized Resource Selection based on Ontology

官上 大輔*¹
Daisuke Kanjo

門林 理恵子*¹
Rieko Kadobayashi

*¹通信総合研究所 けいはんな情報通信融合研究センター

Keihanna Human Info-Communication Research Center, Communications Research Laboratory

In this paper, we propose a framework for personalized resource selection for using any applications accessed from web with any information terminals(PC, PDA, Cellular phone, etc). This framework builds an ontology which reflect individual user's knowledge or interests, and use it to select resources appropriate to user. Since ontology is written by very standard format(RDF/RDF schema and OWL), it can be shared and re-used between applications.

1. はじめに

近年、ネットワーク上に莫大な数のリソースが存在するようになり、ユーザが要求するリソースを特定することが困難になっている。それにともなって、ユーザの関心や知識に応じて適切にリソースを選択することへの要求が強くなっている。この要求に対し、ユーザの行った操作や、実際に選択したリソース/ウェブページのリンクなどの履歴を、ユーザのプロファイルとみなし、それに基づいてリソースを選択する試みが存在する [1, 3, 5]。しかし、プロファイルがアプリケーションに固有であるなど、利用環境が固定されている、あるいは考慮されていないことが多い。その結果、アプリケーション毎にプロファイル獲得の過程が必要となる、あるいは、適応の度合いが異なるなど、といった問題が生じる。

これに対し、ユーザがアプリケーションを利用する際に用いる端末の側にプロファイルを用意することも考えられるが、最近では、ユーザが各種のアプリケーションを利用する際に用いる端末は、PC, PDA, あるいは携帯電話など多岐に渡り、唯一に定めることはできない。したがって、この手法でも、アプリケーションの場合と同様に、個々の端末ごとに、プロファイル獲得の過程が必要となる、適応度が異なる、といった問題が生じる可能性がある。

そこで本研究では、ネットワークに接続されたアプリケーションを不特定の端末から利用し、適応的にリソースを選択することを考えた時に、対象とするアプリケーションと、利用する端末に関する任意性を実現することを目的とする。提案するフレームワークでは、リソースが提示された際に、ユーザが構築する知識の体系をオントロジーによって表現し、それをアプリケーションや端末の間で共有する手段を与えることで、アプリケーションと端末の任意性を実現する。

2. ユーザオントロジー

一般に、ユーザは獲得した情報を整理し、知識の体系を構築していると考えられる。システムがリソースを提示した場合、ユーザは、そのリソースの表す情報を理解し、既に提示されたリソースとの間で共通する要素の抽出や、抽出した共通の要素に基づいて分類を行うなど、リソースに関する知識の体系を構築していると考えられる。この体系は、そのユーザなりの知識や関心、ものごとを理解する際の視点などを表現していると考えられ、ある種のオントロジーとみなすことができる。

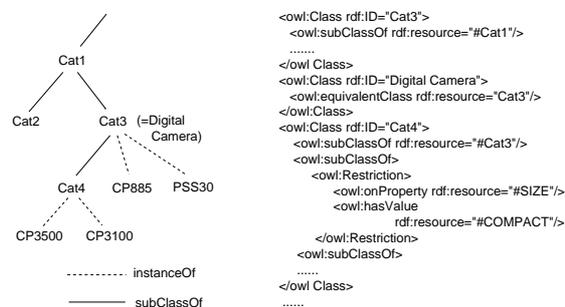


図 1: 例 : ユーザオントロジー

本研究では、このオントロジーをユーザオントロジーと呼び、ユーザが各種のアプリケーションを利用している間にシステムが自動的に構築する。構築したユーザオントロジーは、リソース選択をする際に、当該のユーザのプロファイルとして利用される。ユーザオントロジーは、そのユーザの知識などを表すので、それに基づいてリソースを選択することで、それぞれのユーザの知識や関心などを反映したリソースを適応的に選択できると考えられる。

図 1 に、カメラに関する情報が提示された際に構築されるユーザオントロジーの例を示す。ユーザオントロジーは、カテゴリ (Cat1~4) とリソース (CP3100, CP3500 など、カメラの機種名) が階層化されたツリーで表現される。カテゴリは、そこに属するリソースやカテゴリに関する制約を示す。リソースは、属しているカテゴリの制約を満足する。

図 1 の右は、ユーザオントロジーを RDF/RDF スキーマ, OWL によって記述したものであり、カテゴリはクラスに、リソースはインスタンスに相当する。リソースが持つ属性は Property 要素で、カテゴリが示す制約は Restriction 要素によって記述される。図の Cat4 は、要素 SIZE が COMPACT であるというこのカテゴリに固有の制約と、Cat3 が持っている制約を継承して持っており、そこに属する CP3500, CP3100 は、それらの制約を満たす Property 要素を持ったリソースである。提案するフレームワークでは、このようにユーザオントロジーはツリーのかたちで表現することができ、RDF や OWL によって記述される。

図のようなユーザオントロジーが構築された場合、ユーザは、

連絡先: e-mail:{kanjo,rieke}@crl.go.jp

デジタルカメラを示すカテゴリ (Cat3) が存在し、さらに大きさに従って細分化され、コンパクトというカテゴリ (Cat4) が存在する、という知識を持っているとみなせる。リソース選択の際に、Cat4 にカテゴライズされるリソースを選ぶことで、ユーザの「コンパクトという分類が存在する」という知識を反映できる。

2.1 ユーザオントロジーとリソースの記述

3.2 節で述べるように、ユーザオントロジーは、リソースが持つ属性などに基づいて構築される。また、3.3 節で述べるように、リソースの選択は、リソースが持っている属性などに従って行われる。よって、リソースの属性や構成要素、リソース間の階層関係などを、どのように記述するか、という点が重要となる。その記述が、アプリケーションごとに異なると、構築されるユーザオントロジーに矛盾が生じたり、一貫性を持たないものとなる可能性がある。

そこで提案フレームワークでは、アプリケーションが持つリソースに共通の仕様を与えるために、RDF や OWL などのウェブ上のリソースにメタデータを付与することを目的として作成された言語による記述を仮定する。これにより、異なったアプリケーションのリソースでも一貫した記述が保証され、ユーザオントロジーの構築やリソース選択における統一的な処理が実現できる。また、ユーザオントロジーは、これらの言語で記述されたリソースに基づいて構築されるので、図 1 で示したようにユーザオントロジー自身も RDF や OWL を用いて記述できる。その結果、アプリケーションや端末の間での共有が可能となる。

3. フレームワーク

図 2 に提案するフレームワークの構成を示す。2.1 節で記したように、リソースは、RDF や OWL によって記述されていることを仮定する。

ユーザは、任意の端末から、ネットワーク上の各種のアプリケーションを利用する。利用が始まると、ユーザオントロジーサーバは、当該ユーザのユーザオントロジーを端末のリソースセクタへと転送する。リソースセクタは、ユーザオントロジーを、当該ユーザのプロファイルとして利用してリソースを選択する。リソースセクタ内のユーザオントロジーマネージャは、選択されたリソースなどに基づいてユーザオントロジーを構築する。構築されたユーザオントロジーは、次に提示されるリソースの決定に用いられる。利用を終えると、ユーザオントロジーサーバは、ユーザオントロジーを端末のリソースセクタから受け取り、次の利用まで保存する。

このフレームワークでは、アプリケーションと端末の任意性を実現するために、システムの利用時に、ユーザのプロファイルとして用いるユーザオントロジーを、ユーザオントロジーサーバを介して端末に移動し、共有するための枠組が存在する。また、ユーザオントロジーマネージャにおける、ユーザがアプリケーションを利用している間にユーザオントロジー構築の枠組と、リソースセクタにおけるユーザオントロジーに基づくリソース選択の枠組が存在する。

以下、それぞれの枠組について述べる。

3.1 ユーザオントロジーの共有

ユーザが利用するアプリケーションや端末は常に同じであるとは限らない。しかし、それらについて、それぞれ別個のユーザオントロジーを設けると、そのオントロジーを構築するための過程が個々に必要となるといった問題が生じる。また、よく利用されるアプリケーションや端末では適応度が上がるが、

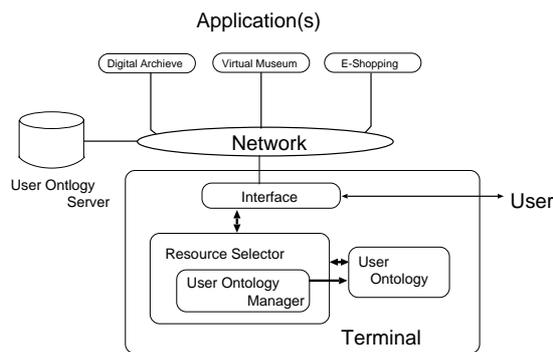


図 2: 提案フレームワークの構成

そうでない端末では適応度が低いままになる、といった問題も生じる。したがって、適応的なリソース選択を行おうという時に、アプリケーションと端末の任意性を実現するためには、プロファイルとして用いるユーザオントロジーを共有するための仕組みが重要となる。

そこで、提案するフレームワークでは、実際にアプリケーションが利用される時のみ、ユーザオントロジーをユーザオントロジーサーバから端末へと転送し、それをユーザオントロジーサーバでリソースの選択を行い、利用されていない時には、ユーザオントロジーサーバでユーザオントロジーを管理する、という手法を採る。こうすることで、一つの端末から、同時に複数のアプリケーションを利用するような場合にも対応できる。

常に同じ個所にユーザオントロジーを置き、そこを介してリソースの選択を行うことも考えられるが、この場合、対象のアプリケーションと端末、それにユーザオントロジーの存在する場所の 3 個所でデータのやりとりが生じるのに対し、提案手法では、一旦、ユーザオントロジーを端末に移動すれば、端末とアプリケーションの間だけでデータをやりとりすれば良い、という有利な点もある。さらに、利用している端末以外の場所にユーザオントロジーが存在する時には暗号化しておく、といった処理を行うことで、プライベートな情報であるユーザオントロジーに関して、セキュリティなどを考慮することも可能となる。

上記を実現するために、ユーザオントロジーサーバは、アプリケーションや端末と同じネットワークに接続され、アプリケーション利用の際にユーザを特定し、そのユーザのユーザオントロジーを端末へと配するサーバの役割を担う。そのために、ユーザ ID とユーザオントロジーを対で管理し、1) 利用開始時に、リソースセクタからユーザ ID を受け取り、該当するユーザのユーザオントロジーを送信する、2) 利用終了後、更新されたユーザオントロジーを受け取り、次に利用されるまでストレージする、という 2 つの機能を有する。

3.2 ユーザオントロジーの構築

ユーザオントロジーは、リソースセクタで、以下の操作に従って構築される。

1. 既存カテゴリへのカテゴライズ : 提示されたリソースを、既存のカテゴリの中でもっとも制約の厳しいカテゴリへカテゴライズする。カテゴライズの可否は、リソースの属性 (Property 要素) が制約を満足するかで判断する。

既存のカテゴリ Cat1 にリソース r1 がカテゴライズされた状態を、図 3(a) に示す。

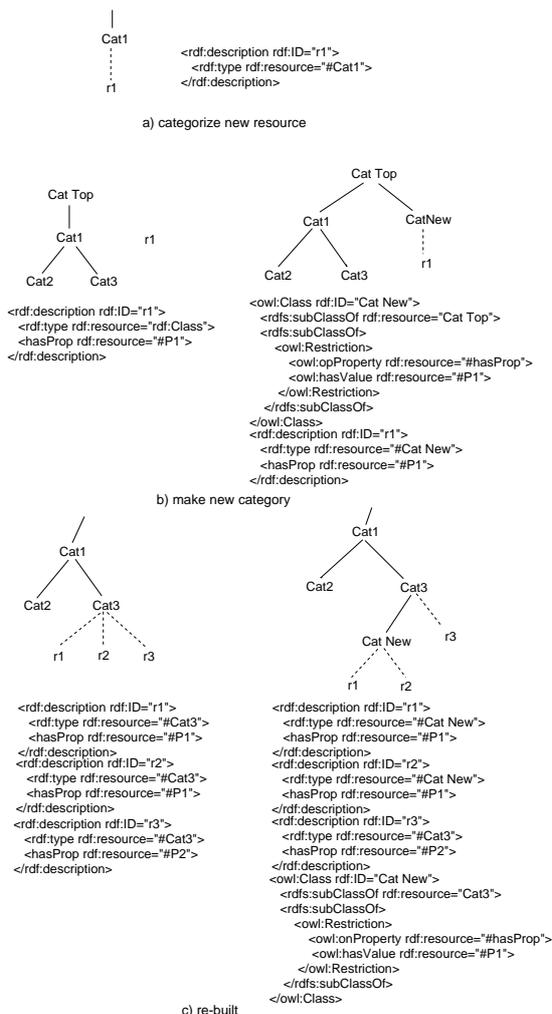


図 3: オントロジーの構築

2. 新規カテゴリの作成 : 1 が達成されない場合, ユーザオントロジーのトップカテゴリの下に, 新しいカテゴリを作成し, そこへリソースをカテゴリ化する. 新しいカテゴリは, そこへカテゴリ化されたリソースに固有の属性を制約として持つ.

この様子を図 3(b) に示す. リソース r1 が既存のカテゴリ Cat1~3 にカテゴリできない時に (図左), 新しいカテゴリ CatNew を作成し, そこにリソースがカテゴリ化される (図右).

3. カテゴリの再編成 : 1, あるいは 2 の操作によって, 同一のカテゴリに 3 つ以上のリソース, もしくはカテゴリが存在するようになった場合, そのリソースやカテゴリに共通の属性を, Restriction 要素として制約に持つ新規カテゴリを作成し, カテゴリの再編成を行う.

この様子を図 3(c) に示す. 1 の操作によって Cat3 にリソース r1~r3 がカテゴリ化された時に (図左), 新たなカテゴリ CatNew が作成され, そこにリソースがカテゴリ化される (図右).

3.3 リソースの選択

リソースの選択は, リソースセクタにおいて, 以下の 3 つのストラテジーに従って行われる.

類推的ストラテジー : 既に提示されたリソースと, 同じ機能や構成要素を持った対象を表すリソースを選択するストラテジー [2]. リソースの Property 要素が, ユーザオントロジーの構築の際に (3.2 節参照), 直前に変化の生じたカテゴリの制約 (Restriction 要素) を満足するようなリソースを選択する. ユーザにとって既知のリソースと新たに提示するリソースの間の類似性を示すことで, 効果的な理解を促すことを試みている.

リソースが唯一に定まらない場合, 持っている属性 (Property 要素) の少ないリソースが選択される.

対比的ストラテジー : 既に提示されたリソースと共通の機能や, 構成要素を持つ一方で, 異なった要素も持った対象を示すリソースを選択するストラテジー [2]. リソース間の差異を示すことで, より効果的な理解を促すことを目的としたストラテジー.

このストラテジーでは, ユーザオントロジーの構築が行われる際に (3.2 節参照), そのリソースが提示されることで, 直前に変化の生じたカテゴリの下位に, 新たに別のカテゴリが作成されるような Property 要素を持つリソースが選択される. また, そのようなリソースが存在しない場合, 変化のあったカテゴリの制約の一部のみを満足するような Property 要素を持つリソースを選択する.

リソースが唯一に定まらない場合, 持っている属性の多いリソースが選択される.

順位付けストラテジー : 上記の 2 つのどちらかのストラテジーに従い, 結果を順位付けて, 候補となるリソースのリストを作成する.

3.4 実行例

本節では, 以下の 2 つのアプリケーションによる実行例を示す. ここでは, ヴァーチャルミュージアムを訪れて, 図 4 の (a) と (b) のようなユーザオントロジーが構築されたユーザが, その後, それぞれ e-ショッピングで同じ条件 (画素数 300 万以上, 記録メディアはコンパクトフラッシュ, 光学ズーム 3 倍以上) で検索を行い, 商品を購入するというシナリオを考える.

ヴァーチャルミュージアム : カメラに関する“展示”を行う. 対比的ストラテジーと類推的ストラテジーに従い, カメラに関するリソースを選択し, 機能等に関する解説を, テキストや図などで行う.

e-ショッピング : デジタルカメラの販売サイトを仮定する. 複数の条件で検索を行い, 特定のデジタルカメラを購入する. リソースの選択は, 類推ストラテジーに従う順位付けストラテジーで行われる.

それぞれ, リソースはデジタルカメラなどで, メーカー, レンズ (倍率), 重量などのデータが記述されているものとする.

まず, 提案するフレームワークによる結果を検証するために, 比較対象として「kakaku.com」(<http://www.kakaku.com>) の検索結果を考える. kakaku.com では, 複数の検索条件を入力し, それを満足する対象製品の一覧を出力として得ることができる. kakaku.com で, 前述の条件で検索を行うと, 表 1 右の

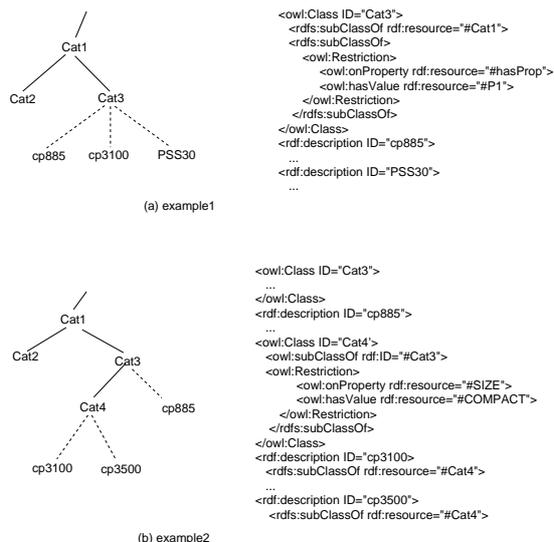


図 4: 構築されたユーザオントロジーの例

欄のような、44 機種を列挙した結果が得られる。この結果は、ユーザオントロジーが構築されていない場合の結果と考えることができる。

次に、本フレームワークによる結果を考える。比較を容易にするために、本研究では kakaku.com と同じ 44 機種を対象としたデータを用い、図 4 の (a) と (b) のそれぞれのユーザオントロジーが構築されている場合を考える。

44 の機種のうち、CP3500,CP3100,Opt330GS,Opt33L,IXTD400,CPSQ の 6 機種は Property 要素 SIZE の値が COMPACT であり、(b) のユーザオントロジーが構築されている場合、Cat4 の下にカテゴリ化される。また、それ以外の機種は Cat3 の下にカテゴリ化される。e-ショッピングが採用している順位付け戦略では、直前に変化のあったカテゴリにカテゴリ化されるリソースの方が優先されるため、その結果は、表 1、左の欄のように、6 機種が先頭になる。この結果は、次に 2 つの事項を示していると考えられる。

- kakaku.com による結果、すなわち、ユーザオントロジーが構築されていない場合と比較すると、ユーザの知識を反映したリソースが選択されていること。
- 他のアプリケーション（ヴァーチャルミュージアム）で構築されたユーザオントロジーに基づいて、適応的なリソースが選択が行われていること。

このように、提案するフレームワークでは、ユーザオントロジーを異なるアプリケーションで共有し、適応的にリソースの選択ができることを示している。

一方、(a) のオントロジーが構築されている場合、適当なカテゴリが存在しないため、44 機種全てが Cat3 にカテゴリ化されてしまい、kakaku.com と同じ結果となる。これは、異なるユーザオントロジーによって、異なる結果が得られることを示すと同時に、適切なユーザオントロジーが構築されていない場合、妥当な結果が得られないことを示している。

4. おわりに

本研究では、ユーザの関心や知識に応じてリソース選択を行うことを考えた場合に、対象とするアプリケーションと、利

表 1: 検索結果

提案フレームワーク	kakaku.com	
CP3500	175g	CP885 225g
CP3100	150g	CP3500 175g
Opt330GS	180g	CP3100 150g
Opt33L	180g	Opt330GS 180g
IXTD400	185g	Opt33L 180g
CPSQ	180g	PSA70 -
CP885	225g	PSS30 260g
PSS30	260g	QV4000 355g
...
Opt430	240g	Opt330 205g
RDCi700G	450g	Opt430 240g
PSA70	-	RDCi700G 450g

用する端末の任意性を実現するためのフレームワークを提案した。ユーザの知識体系（オントロジー）を RDF や OWL で記述し、それをプロファイルとして、各種のアプリケーション、および端末から共有・利用を可能とすることで、任意性を実現している。

また、本稿では、実際に異なったアプリケーションでユーザオントロジーを共有し、適応が実現されることを示した。一方、現在システムは試作段階であり、端末の任意性を実際に検証するには至っていない。今後は、この点について検証をするとともに、リソースを選択するための戦略の拡充などについても考えていきたい。

参考文献

- [1] Cooley,R.Tan,P.N.,Srivastava,J, WebSIFT: The Web site information filter system, In Proc. of Web Usage Analysis and User Profiling Workshop(1999)
- [2] 官上大輔, 門林理恵子, オントロジーに基づくオブジェクトの適応的選択, 信学技報 Vol.102, No.603(2003)
- [3] Kurki,T.,Jokela,S.,Sulonen,R.,Turpeinen,M., Agents in delivering personalized content based on semantic metadata, In Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace, pp84-93(1999)
- [4] Dean,M.,Connolly,D.,Harmelen,F.,Hendler,J., Horrocks,I.,McGuinness,D.,Schneider,P.,Stein,L., OWL Web Ontology Language 1.0 Reference, <http://w3.org/TR/owl-ref/>
- [5] Pretschner,A.,Gauch,S., Ontology Based Personalized Search, In Proc. of 11th Intl. Conf. on Tools with Artificial Intelligence(1999)
- [5] Beckett,D.,RDF/SML Syntax Specification, <http://www.w3c.org/TR/rdf-syntax-grammar/>
- [6] Brickley,D.,Guha,R.V., RDF Vocabulary Description Language 1.0:RDF Schema, <http://www.w3.org/TR/rdf-schema/>