

エージェントの得点付け手法により個別化を行う プライベート仮想ライブラリ PVL の提案と評価

上田 真由美[†] 上 島 紳 一[†]

インターネット上の膨大な情報資源を個別化して利用するための枠組みが重要視されている。本稿では、利用者がインターネット上に存在する情報資源を個人の好みで利用するための枠組みとして、プライベート仮想ライブラリ PVL を提案する。PVL では、インターネット上の情報を本に見立てて本棚を構築する本棚メタファを用いている。PVL は、複数のアクセスエージェントとメディアータによって実現するメタ検索エンジン機構と、XML 形式の情報ベースで構成されている。また、PVL の個別化機構は、利用者の代表的な対話的操作によって行うアクセスエージェントに対する得点付けに基づいている。この得点をフィードバック利用することで、利用者は意識することなく、段階的にシステムの個別化を進めることができる。本手法の有効性を検証するため、プロトタイプシステムを構築し評価実験を行った。実験結果により、既存の検索エンジンと比較して、格納に関する適合率が約 34.1%、表示に関する適合率が約 12.8% 上昇することを確認した。

Proposal and Evaluation of a Private Virtual Library PVL with Personalization Mechanism Based on Agent Scoring

MAYUMI UEDA[†] and SHINICHI UESHIMA[†]

Recently, personalization has been one of central research issues for utilizing large volume of WEB contents over the Internet for personal use. In this paper, the authors propose a Private Virtual Library PVL, which is a combination of meta search engine and information repositories using BookShelf metaphor. PVL has an embedded mechanism that maps users' interactive operations to scores of access agents to information sources, and users can personalize his BookShelf by using PVL incrementally. We construct prototype system, and evaluate our system with respect to the movement of two kinds of precision factors, which we define for user's browsing and saving actions. The numerical results show that the precision factor of saving increases by 34.1%, while precision factor of browsing increases by 12.8%, in comparison to the existing search engines.

1. はじめに

近年の高度情報化にともない、様々な情報資源がインターネット上に構築されている。情報量の急激な増加により、個人にとって有効な情報を入手することが困難となり、インターネット上の情報資源を個別化して利用するための枠組みに関する研究がさかんである^{1),2)}。

本稿では、利用者がインターネット上に存在する情報資源を個人の好みで利用するための枠組みとして、プライベート仮想ライブラリ PVL を提案する^{3),4)}。PVL では、インターネット上の情報を本に見立てて本棚を構築する、“本棚”のメタファを用いている。つ

まり、現実の本棚と仮想の本棚の利点を融合した夢の本棚としての実現を目指している。

このような仮想ライブラリの構築には、検索機構が重要である。Yahoo! に代表されるディレクトリ型の検索エンジンは、検索エンジン構築者の手作業を必要とするため、網羅性は低いですが、提供情報の信頼性は高い。これに対し、Infoseek などのロボット型の検索エンジンでは、人手に頼らず、プログラムが自動的に収集しているため、情報の鮮度が高く、網羅性も高い⁵⁾。しかし、各検索エンジンが独自の検索モデルやデータ整理法を用いるため、同一の検索キーワードに対して、異なる検索結果が提供される⁶⁾。そこで、PVL では既存の検索エンジンが提供する情報を統合利用するメタ検索機構を構築する。メタ検索エンジン機構では、複数の検索エンジンを統合利用することによって、(1) 各検索エンジンの偏りを生かすことで網羅性を高め、

[†] 関西大学大学院総合情報学研究科
Graduate School of Informatics, Kansai University

(2) 重複性を生かすことで提供情報の信頼性を高めることを目標としている。また、各検索エンジンは異なる形式で情報提供を行っているため、差異を吸収して蓄積可能な情報ベースが必要である。そこで、PVLでは情報ベースとして、XMLを用いた共通形式の本棚を個人ごとに提供し、各検索エンジンが提供する情報を格納する。

PVLは、エージェントを用いたメタ検索エンジン機構とXML形式の情報ベースで構成されている。メタ検索エンジン機構では、インターネット上の複数の検索エンジンを利用して、問合せを行い結果を集約・個別化し、利用者ごとに順序付けられた情報を提供する。

この個別化の過程では、利用者によるWEBサイトのブラウジング中の選択や保存といった代表的な対話的操作を得点化しており、システムがこれらをフィードバック利用することで、段階的にライブラリが個別化されていく仕組みを持つ。この仕組みによって、利用者は個人の好みの本棚を意識せずに構築することができる。

既存の検索エンジンでは、独自の得点付けを用いている。それらは、利用者へ提供する情報は一意に決定する⁵⁾。また、他の多くの個別化サービスを提供しているWEBサイトでは、あらかじめ、利用者が個人の興味を登録しておく必要があることが多い。この登録形式による利用者の興味獲得は、正確な利用者の興味を獲得することが可能であるが、利用を繰り返すうえで興味の変化があれば、登録内容を変更する必要があり、利用者への負担が大きいと考えられる^{7),8)}。

PVLにより

- 対話的操作で得点付けを行い個別化するので、利用者は意識せずに個別化した本棚を構築し利用できる。さらに、利用時の対話的操作によって個人の好みを獲得するため、利用者の好みの変化をつねに更新していくことが可能である、
 - 検索エンジンごとに作成したアクセスエージェントを得点付けしているため、個別化機能を持たない情報資源に対しても、自分の嗜好を反映したライブラリを構築できる、
 - 情報ベースの構築にXMLを利用することで、様々な形式のデータを扱うことができ、容易に並べ替えや選択が可能になる、
- などの利点を持つ。

2章では本棚メタファについて述べ、3章で、本システムの基本構成、4章で、エージェントの得点付け手法について述べる。5章でプロトタイプシステムを構築し、評価実験により本システムの有効性を検証す

る。6章で関連研究との比較を行い、最後に7章で本稿のまとめを行う。

2. 本棚メタファ

我々が考える本棚メタファとは、インターネット上にある自分好みの情報を、自分の本棚にある本のように、好きなときに自由に扱うことができるような仮想の本棚を用いたメタファである。ここでは、次の点に着目している。

【インターネット上の情報を本棚に並べる】

自分の部屋の本棚に好きな本が並んでいるのと同様、PVLの本棚には、インターネット上の好きな情報を格納しておく。利用者はこの仮想ライブラリの利用時に、必要と判断した情報をカテゴリに分類して保存し、再利用することができる。

【手元のない本も本棚に】

特定の本を保存するだけでなく、“あんな本”や“こんな本”といった条件を保存する。これにより、必要なときに手元のない本や欲しい本をインターネット上から収集することができる。

【本の多重分類が可能】

本棚に格納する際、利用者は1つの本を複数のカテゴリに格納することができる。

【欲しい本が欲しい順に】

本棚に保存した情報は、利用者の要求に応じて、保存時に選択したカテゴリ情報を元に、必要なカテゴリを選択して表示する。また、保存した日時によって並べ替えて利用する。

【繰り返し利用による本棚の個別化】

本棚は、利用者個人の視点でインターネット上の情報資源を格納し、ボトムアップに利用者のもとに構築する。利用者が利用を重ねるたびに、より利用者の嗜好に近づいてゆく。

【本のようにページをめくる】

利用者は目的に応じて、3D形式のGUIを用いて、実際の本のようにページをめくりながらブラウジングする⁹⁾。利用者は本棚の本を用いて、必要なときに、必要な情報をブラウジングすることができる。

PVLは、現実の本棚と仮想の本棚の利点を融合した個人向け仮想ライブラリである。次章では、これらのPVLの要件を満たすシステムの構成について述べる。

3. 基本構成

PVLは、協調的に働くエージェント群によるメタ検索エンジン機構と本を格納する情報ベースから構成

される。

3.1 エージェント群

メタ検索エンジン機構の実現には、各検索エンジンごとに定義したアクセスエージェントと、それらを統合するメディアータを用いている。また、利用者ごとの個別化のための情報は利用者プロフィールに蓄積する。

図 1 に本システムの基本構成とデータの流れを示す(データの流れについては 4.1 節で述べる)。図 1 中 “IS” は、インターネット上に存在する情報資源、 SE_1, SE_2, \dots は既存の検索エンジンを示し、網掛け部分が PVL を示す。

【アクセスエージェント】

PVL では、既存の検索エンジンごとにアクセスエージェントを備えている。これらのアクセスエージェントは、各検索エンジンが提供する情報を PVL で利用する形式に統一するための、ラッパーの働きを備えている。

すなわち、アクセスエージェントは、利用者が入力したキーワードのブール表現を URL に変換する。ただし、変換例は 5.1 節の Example3 に示す。また、反対に情報資源から収集した情報を HTML のタグを利用し、各 WEB ページのタイトルと URL 部分を抜き出し、PVL で利用する形式に統一する。抽出した情報は、すべてメディアータに返す。

【メディアータ】

メディアータは、アクセスエージェントやプロフィールエージェントなどエージェント群を統合する働きをする。ここでは、各アクセスエージェントからの結果を集約し、利用者プロフィールから得た各アクセスエージェントの得点をもとに個別化処理を行う(4 章で述べる)。その後、GUI に提供する。

【プロフィール・本棚エージェント】

プロフィールエージェントは、個別化に用いる利用者プロフィールの参照・更新を行う。また、本棚エージェント(図中では BookShelf Agent)は、本棚への格納・並べ替えを行う。これらのエージェントは、メディアータの要求によって処理を行う。

本システムでは、エージェント群がメディアータを中心に働いている。既存の検索エンジンを利用したメタ検索エンジンを構築し、クライアントに情報を集めメディアータで個別化処理を行う。このため、PVL ではサーバ上で個別化のための得点付けを行っていない情報を、個別化して利用することが可能になる。

【GUI】

利用者は GUI を介してメタ検索エンジン機構の利用や本棚の操作を行う。

PVL では、利用者の目的や嗜好に応じて GUI を切り替えることを目指して、2 次元ブラウザと、試験的に作成した本のページをめくる操作を実現した 3 次元ブラウザの 2 種類の GUI を作成している⁹⁾。

3.2 本棚の構成

個別化処理を行って利用者に提供された情報は、利用者の判断によって、本棚に格納する。本棚への格納時には、システムが次の DTD (Data Type Definition) に示す XML 形式にタグ付けを行って格納する。

Example1: 本棚ファイルの DTD

```
<!DOCTYPE 本棚 [<!--本棚の構造-->
<!ELEMENT 本棚 (Book+)>
<!ELEMENT Book
    (category*, keyword*, title, url, date)>
<!ELEMENT category(#PCDATA)>
<!ELEMENT keyword(#PCDATA)>
<!ELEMENT title(#PCDATA)>
<!ELEMENT url(#PCDATA)>
<!ELEMENT date(#PCDATA)>
]>
```

XML 形式を用いることにより、本棚を利用する場合、意味的な情報を利用したり、カテゴリを限定して本棚を表示したり、格納日時により並べ替えることができる利点を持っている。

図 3 に PVL の GUI を示す。右の (A), (B) 部分がそれぞれ 1 つの本を示す。本棚には、複数の本が格納されている。各本にはそれぞれカテゴリ、キーワード、タイトル、URL、格納日時の情報が付加されている。(A) は特定の WEB サイトを格納しているが、(B) は条件を格納しているため、必要ときにインターネット上から収集し、結果の中から興味にあう情報を選択することが可能である。

4. エージェントの得点付け手法

本章では、個別化のためのエージェントの得点付け手法について述べる。

4.1 個別化処理とデータの流れ

3 章で構成要素を述べたように、本システムはメタ検索エンジン機構を構成するエージェント群が協調的に働き、カテゴリごとにアクセスエージェントの得点を用いて個別化を行う方法をとっている。

図 1 を用いて各エージェントの動きとデータの流れについて述べる。図の矢印についた A, B, C, D はそれぞれカテゴリ、キーワード(質問条件)、エージェントの得点、問合せ結果を示しており、エージェントの動きにともなうデータの流れを示す。

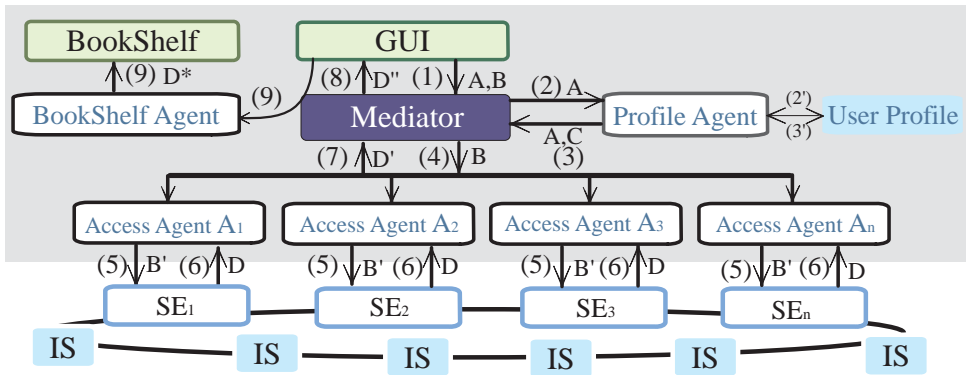


図1 PVLの基本構成とデータの流れ
Fig. 1 Basic architecture and data flow.

以下では、カテゴリ = C_i ，キーワード = “ k_a ” and “ k_b ” を例に述べる．カテゴリ C_i は利用時に利用者が選択したカテゴリで、 k_a, k_b は、それぞれ利用時に利用者が任意に入力した検索キーワードである．利用するアクセスエージェントは n 個とし、 A_j ($j = 1, 2, \dots, n$) と書く．検索エンジンからの問合せ結果は R_j とし、各検索エンジンが返す HTML 形式のデータである．また、カテゴリ C_i に対する各アクセスエージェントの得点は $Score_{ij}$ とする ($Score_{ij}$ は正の実数)．

【利用者による操作】

利用者は、カテゴリを選択し、キーワード（質問条件）を入力する（図1矢印1）．

【矢印(1)】 $A: C_i, B: “k_a”$ and “ $k_b”$

【利用者プロフィールの参照】

メディエータは、矢印(1)で利用者インタフェースからカテゴリとキーワードを受け取ると、プロフィールエージェントに各アクセスエージェントの得点の計算要求を出す（図1矢印2）．

【矢印(2)】 $A: C_i$

【アクセスエージェントの動き】

メディエータは、プロフィールエージェントから計算結果を受け取ると（図1矢印3）、各アクセスエージェントに対して検索要求を出す（図1矢印4）．

【矢印(3)】 $A: C_i, C: Score_{ij}$

ただし、アクセスエージェントの得点 $Score_{ij}$ の計算法は4.2節で述べる．

【矢印(4)】 $B: “k_a”$ and “ $k_b”$

検索要求を受け取った各アクセスエージェントは、3.1節で述べたように、キーワードで構成される検索要求をURLに変換し、既存の検索エンジンを用いてインターネット上の検索を行う（図1矢印5）．

【矢印(5)】 $B': url_j$

ただし、 url_j はアクセスエージェント A_j が変換した

検索エンジンに対応するURLを表す．変換例は5.1節のExample3に示す．

各アクセスエージェントは、インターネット上から収集した結果 (R_j) を受け取ると（図1矢印6）、HTMLのタグを用いてPVLで利用する形式 (R'_j) に変換する．ただし、 R'_j は順序付きリストで、 $R'_j = (R'_j(1), R'_j(2), \dots, R'_j(m_j))$ ($j = 1, 2, \dots, n$) とし、 m_j は各 R'_j に含まれるURLの個数を示す．

【矢印(6)】 $D: R_j$

【メディエータによる処理】

メディエータは、各アクセスエージェントからPVLで利用する形式に統一した結果 (R'_j) を受け取ると（図1矢印7）、先に計算しておいたアクセスエージェントの得点 ($Score_{ij}$) を利用して、集約・並べ替えを行い順序付きリスト ($Final Result FR$) を作成する．集約・並べ替えの方法は4.3節で述べる．

【矢印(7)】 $D': R'_j$

【利用者への提供】

個別化された結果 (FR) は、図2のように表形式でGUIを介して利用者に提供する（図1矢印8）．

【矢印(8)】 $D'': FR$

FR は順序付きリストで、 $FR = (FR(1), FR(2), \dots, FR(m'))$ となる．ここで m' は FR の長さを示し、 $0 \leq m' \leq \sum_{j=1}^n m_j$ を満たす．また、 $FR(k')$ は各アク

セスエージェントがPVLで利用する形式に変換した R'_j 内の情報を含む $R_R(k')$ と、各結果に対する得点 $RV_R(k')$ を含む．

利用者への提示時には、既存の検索エンジンから得たページのタイトルとURLだけでなく、本手法によって付けられた各ページの得点と、そのページを検索結果に含む検索エンジンの名前を提供している．

さらに、提供された情報をブラウジングした後、利

N.	Title	URL	Ranking Val.	agent_name
1	Japan XML User Group Web Pa...	http://www....	251.285714...	excite&infos...
2	Extensible Markup Language (XML...	http://www....	72.0	lycos
3	XML Consortium	http://www....	53.05	excite&infos...
4	Fuji Xerox Information Systems Co.,Ltd...	http://www.f...	31.3426651...	excite&infos...
5	XML eXpert eXchange	http://www....	26.3502262...	excite&infos...
6	拡張可能なマーク付け言語 (XML...	http://www.f...	18.9	excite&yahoo
7	ZDNetエンタープライズ：トップページ	http://www....	18.0	lycos
8	developerWorks：XML	http://www....	16.2727272...	infoseek&lyc...
9	XMLマスター認定制度：ホーム...	http://www....	12.6802114...	excite&infos...
10	sf's XML Page	http://www....	11.8663594...	excite&infos...
11	developerWorks：XML	http://www.i...	11.8125	excite&yahoo
12	XML PRESS	http://www....	11.0972850...	excite&lycos...
13	XML Magazine	http://www....	10.7944272...	excite&lycos...
14	XML	http://db-w....	10.3380488...	excite&infos...
15	たのしいXML:XML/XML...	http://www....	9.47497472...	excite&infos...
16	[TOIC XML/SGML] XML/f...	http://www.t...	9.09286656...	excite&infos...

図 2 本アルゴリズムを用いて個別化を行った際の利用者 A, B への情報提供例

Fig. 2 Top ranked WEB pages for user A and user B.

図 3 プロトタイプシステムの画面例

Fig. 3 Overview of prototype system.

利用者の判断で本棚に格納できる (図 1 矢印 9)。

【矢印 (8)】D* : Book

ただし、各 Book は 3.2 節で述べた DTD に従って XML 形式にタグ付けを行ったものである。

4.2 得点付け法と計算法

本節では、アクセスエージェントの得点付け法と計算法について述べる。

本システムでは、利用者のブラウジングにおける対話的操作によって付けられたアクセスエージェントの得点を利用して、問合せ結果に対する得点付けを行っている。

【アクセスエージェントの得点付け】

本システムでは、次の 2 つの利用者による対話的操作 (操作 1) (操作 2) をアクセスエージェントの得点と対応付けている。

(操作 1) 図 2 の形式で提供された問合せ結果の中から興味のある情報を選択し、実データを表示させる。

(操作 2) 図 3 左上のフレームで実データをブラウジング後、必要と判断し本棚に格納する。

この操作を行うたびに、利用者プロファイルに、[カテゴリ、キーワード、アクセスエージェントの得点、保存した日時] の形式で保存する。

利用者は繰り返しシステムを利用することで、意識することなく、アクセスエージェントに対し得点付けを行う。

通常、利用者の興味を自動的に獲得する手法として、(操作2)が有効であると考えられる。しかし、利用者がインターネット上から得た情報を本棚に格納する際、明確な目標を持ち検索を行い、本棚に格納する場合だけでなく、漠然とした目標から検索を行い、検索過程において必要な情報を発見し本棚に格納することが多い。そこで本手法では(操作2)だけでなく、必要な情報を見つける過程も利用者の興味獲得に有効な要素であると考え(操作1)(操作2)を組み合わせて興味獲得に利用する(操作1)(操作2)により自動的に個人の興味を獲得しフィードバック利用するため、利用者に負担をかけずに、個人の興味を反映させた情報提供が可能になる。

【アクセスエージェントの得点の計算法】

利用者プロフィールに記述されたアクセスエージェントの得点は、カテゴリごとに計算し個別化に利用する。先に述べた利用者の多様な検索状態(明確な目標・漠然とした目標)に対応するため、アクセスエージェントの得点を、過去の利用者の操作によって付与された得点の単純和によって算出する。なお(操作1)(操作2)の得点の値の決定については、5.2.2項で述べる。つまり、カテゴリ C_i におけるアクセスエージェント A_j の得点 $Score_{ij}$ は、

$$Score_{ij} = \sum_{k=1}^n W_{ij}^k \quad (1)$$

と計算する。上式で、 W_{ij}^k は第 k 回目の操作で得られる、カテゴリ C_i に対するアクセスエージェント A_j の得点を示す。式(1)で n を変動させると、過去 h 回、過去 h 日とフィードバックするアクセスエージェントの得点の範囲を決定することができる。この n の範囲は、利用者が個人の目的や嗜好に応じて、自由に決定可能であることが望ましい。ただし、5.2節では n の値を固定し、過去すべての W_{ij} の値を用いて実験を行った。

【各データ (url_j) に対する得点の計算法】

PVLでは各アクセスエージェントが収集した情報の順序と、各アクセスエージェントの得点とを用いて、問合せ結果に対する得点付けを以下の計算により行う。つまり、カテゴリ C_i におけるアクセスエージェント A_j が収集した第 k 番目のデータに対する得点 *Ranking Value* $RV_j(k)$ は、

$$RV_j(k) = Score_{ij} \times \frac{1}{k} \quad (2)$$

により計算する。 $RV_j(k)$ は正の実数である。

各検索エンジンはアクセスエージェントを通して同等に取り扱い、各アクセスエージェントの優先度は式(1)の $Score_{ij}$ のみによって決定している。また、各検索エンジンの提供順序を利用するため、各アクセスエージェントが収集した順序に対し、同一の単調減少な重み付けを行う。ここでは、そのような関数として簡便のため $1/x$ を用いて重み付けした。

各アクセスエージェントから返された情報に対し、上記の計算で得点付けを行い、結果を集約して並べ替える。ただし、複数のアクセスエージェントから同じ情報が返された場合、得点の合計を、その情報の得点とする。

4.3 マージング手法

表1に、各アクセスエージェントからの問合せ結果のマージングと並べ替えアルゴリズムを示す。利用者の要求により、プロフィールエージェントが利用者プロフィールから計算した各アクセスエージェントの得点 $Score_{ij}$ と、各アクセスエージェントがインターネット上から問合せ結果をPVL形式に統一した R'_j を得ると、アクセスエージェント A_j が収集した情報に対する得点 (*Ranking Value* $RV_j(a)$) の計算を行う(Step(4))。

各アクセスエージェントが収集した情報の中から、得点 $RV_j(a)$ が最大の情報を検索し、 $R'_j(k)$ が *Result R* に含まれていない場合、*R* に $R'_j(k)$ と $RV_j(k)$ を追加する(Step(5), (6))。

$R'_j(k)$ がすでに *R* に含まれる場合、 $R'_j(k) = R_R(i)$ となる i を検索し、 $RV_R(i)$ に $RV_j(k)$ を追加する(Step(5), (8), (9))。

各アクセスエージェントから返された情報 *R* に対し、各情報の得点 RV_R を用いて並べ替えた結果 *FR* を利用者に提供する(Step(11), (12))。

メタ検索エンジンでは、重複データを削除し利用者に提供するものが多い。しかし、各検索エンジンが独自の手法を用いて情報提供を行ったにもかかわらず重複している情報は、利用者が与えた検索キーワードに対して重要な情報であると判断し、本システムではそれぞれの *RV* を加算する。

Example2: 利用者A, Bを用いた実例

図2は、利用者A, Bが同じカテゴリとキーワー

単調減少関数として、線形関数や対数関数なども考えられるが、ここでは簡便のため $1/x$ を用いた。

表 1 問合せ結果のマージングと並べ替えアルゴリズム
Table 1 Merging and sorting algorithm for meta-search engine.

Merging and Sorting for Personalization –for Private View
Start : Result R'_j that is result set retrieved by Agent A_j using search engine SE_j
 $Score_{ij}$ that is score of Agent A_j in category C_i calculated from *Profile*
/* Let n be number of agents, A_1, A_2, \dots, A_n */
Method:
(1) result set $R'_j(a) = \{R'_j(1), R'_j(2), \dots, R'_j(m_j)\}$
(2) for $j = 1, n$ // Agent₁, Agent_n
(3) for $a = 1$ to m_j // $R'_j(1)$ to $R'_j(m_j)$
(4) Compute Ranking Value $RV_j(a)$ for all results
 $RV_j(a) = Score_{ij} \times \frac{1}{a}$
(5) Find max value in $RV_j(a)$
(6) if $R'_j(a) \notin \text{Result } R$
 add $R'_j(a)$ and $RV_j(k)$ to R
/* $R(i) = (R_R(a), RV_R(a))$ */
(7) Go to Step(5).
(8) if $R'_j(a) \in R$
 Find i_0 such that $R'_j(a) \subset R(i_0)$
(9) Replace $RV_R(i) = RV_R(i) + RV_j(a)$
(10) Go to Step(5).
(11) sorting R in the descending order, using RV_R
 into Final Result FR
(12) Output FR that is merging and sorting by RankingValue RV

ドを用いて本システムを用いたときに提供された結果である。ただし、ここでは、カテゴリは“computer”，キーワードは“XML”を用い、4種類のアクセスエージェントを利用した。

図 2 左が利用者 A，右が利用者 B の結果である。

- (1) Start
利用者 A の利用者プロフィールに記述された“computer”カテゴリにおける各アクセスエージェントの得点は、それぞれ $Score_1 = 91$ ， $Score_2 = 52$ ， $Score_3 = 98$ ， $Score_4 = 72$ ，利用者 B は $Score_1 = 94$ ， $Score_2 = 14$ ， $Score_3 = 97$ ， $Score_4 = 61$ とする。ただし、利用者 A，B はそれぞれ 10 回検索を行った後の状態である。
- (2) Step(4)
利用者 A が得た情報に対して、表 1 の Step(4)を用いて各アクセスエージェントが得た情報に対する得点の計算を行う。アクセスエージェント A_1 が得た情報に対して、 $RV_1(1)$ から $RV_1(m_j)$ まで計算を行うと、 $RV_1(1) = S_1 \times \frac{1}{1} = 91 \times \frac{1}{1} = 91$ となり、 $RV_1(a) = 91.0, 45.5, 30.33, \dots, 91 \times \frac{1}{m_j}$ となる (ただし $a = 1, 2, 3, \dots, m_j$ とする)。同様に、アクセスエージェント A_2, A_3, A_4 についても RV の計算を行う。

- (3) Step(5), (6)
この結果、Step(5)により、 $RV_j(a)$ の最大値を検索し、 $RV_3(1) = 98$ が得られる。よって、 $R_3(1)$ を R 内の情報と比較し、同じ情報がないことを確認し、 R に $R_3(1)$ と $RV_3(1)$ を追加する。
- (4) Step(8), (9)
(2) で得た $R_3(1)$ が、すでに R に含まれる場合、 $R_3(1) = R_R(i)$ となる i を検索し、 $RV_R(i)$ に $RV_3(1)$ を加算する。
- (5) Step(11), (12)
(2), (3), (4) の処理をすべての行った後、 R に対して RV_R を用いて並べ替えを行い、 FR を利用者に提供する。□

上記の処理では、利用者 A，B は、同じカテゴリ、キーワードを用いて検索を行っているが、異なる情報が提供される (図 2)。既存の検索エンジンが同じ情報を上位に表示したことと、本システムの利用頻度が少なく (両利用者とも 10 回使用後のデータ)、十分に個別化されていないため、両利用者とも同じ情報が上位に表示されている。

図 2 では、両利用者とも同じ情報が 1 件目に表示されているが、各アクセスエージェントの得点が異なるため、同じ情報に対する得点が異なっている。また、利用者 A (左) で 10 件目に提供された情報を、利用

者 B に対して 16 件目に提供している．利用者 B に 13 件目に提供した情報は，利用者 A には 16 件以内に提供していない．

4.4 特徴

PVL におけるエージェントの得点付けを用いた個別化手法は，以下の特徴を持つ．

- 利用者は繰り返し利用することによって，意識することなくシステムの個別化を進める．
- アクセスエージェントに対する得点を用いて個別化を行うため，サーバ上であらかじめ得点付けされていない情報を個別化して利用することが可能である．すなわち，インターネット上に存在する大量の情報を，利用者側で個別化処理を行い利用することが可能である．
- 利用者ごと，またカテゴリごとに同じアクセスエージェントの得点が異なっている．つまり，同じメタ検索エンジン機構を用いても，利用者の好みの情報が得られる．
- “あんな本” や “こんな本” といった条件を保存し，必要なときにネットワーク上から収集できる．
- 各アクセスエージェントとメディアータが協調して，各々の役割を果たすことで，アクセスエージェントがインターネット上から収集した情報をクライアント側で集約し処理することが可能となる．

5. プロトタイプシステムの実装と評価

本手法の有効性を考察するために，提案手法を用いたプロトタイプシステムを実装し，評価実験を行った．

5.1 プロトタイプシステム

図 3 にプロトタイプシステムの画面例を示す．

利用者への情報提示のため 2 次元 GUI と実際の本の 3D 形式の GUI を作成した．また，本棚には Example1 に示した XML 形式を用いている．

メタ検索エンジン機構を実現するため，既存の検索エンジン (Yahoo!JAPAN⁸⁾，インフォシーク¹⁰⁾，Excite エキサイト¹¹⁾，Lycos Japan¹²⁾) に対応するアクセスエージェントを作成した．各々のアクセスエージェントは，3.1 節で述べたように，キーワードで構成されるブル演算を URL に変換する．以下に，各アクセスエージェントの検索エンジンに対する変換例を示す．

Example3: キーワード = “Java” and “XML”
<http://google.yahoo.co.jp/bin/query?=Java+XML>
<http://www.infoseek.co.jp/Titles?qt=Java+XML&btnchk=1&lk=noframes&qp=0&nh=50>
http://www.excite.co.jp?search.gw?target=web&look=excite_jp&lang=jp&search=Java+

XML&perPage=50 □

利用者は，左下のオペレーションフレームでカテゴリを選択し，キーワードを入力する．ただし，カテゴリは既存の検索エンジンのカテゴリ分類を参考に PVL が提供する．利用者の要求によって，インターネット上から収集した情報は，利用者プロフィールに記述された各アクセスエージェントの得点を利用して個別化を行う．個別化された情報は，図 2 の形式で利用者に提供される．その中から，利用者が興味を持ち選択した情報は，タブを切り替えることで，図 3 左上のように実際のページが表示される．さらに，実際のページをブラウジングし興味ある情報と判断し，本棚 (図 3 右) に保存しておくことが可能である．

この利用者の “選択” 操作で，その情報を収集したアクセスエージェントに対して加点する．また，利用者の本棚への “保存” 操作によって，さらにアクセスエージェントに対して加点する．

本棚の (A) 部分が 1 冊の本で，保存した 1 つのサイトを表している．本棚は，利用者の対話的操作で利用者側に構築され，必要に応じて自由に利用することが可能である．本棚の構築は，3.2 節の Example1 で示した DTD に従って作成している．

図 3 右下は，現在検出中の 3 次元ブラウジングインタフェースである．利用者側での個別化には，利用者インタフェースの切替えが有効である．この GUI は，長年親しんだ本と同様の操作が可能のため直感的に操作でき，多くの利用者に効果的であると考えられる．

実装には主に，Java2 SDK v.1.2.2 を用いた．また，XML パーザーには xml4j を用いた．PVL は Windows2000 と Solaris8 上で稼働を確認している．3 次元 GUI の作成は，Java3D と VRML2.0 で作成した⁹⁾．

5.2 評価実験

5.2.1 評価パラメータ

5.1 節で述べたプロトタイプシステムを用いて，研究室の学生 10 名に PVL を実際に使用してもらい評価実験を行った．次の 2 種類の適合率 α ， β の変化を追跡した．ただし，格納に関する適合率を適合率 α ，表示に関する適合率を適合率 β とし，それぞれ次式で計算を行う．

$$\text{適合率 } \alpha = \frac{\text{本棚に格納した数 } (C)}{\text{検索結果の数 } (A)} \quad (3)$$

$$\text{適合率 } \beta = \frac{\text{実データを表示した数 } (B)}{\text{検索結果の数 } (A)} \quad (4)$$

検索エンジンを利用時に大量の問合せ結果が提供され，その中から有効な情報を探るのが困難である．そこで，上位に有効な情報を表示することに注目し，本

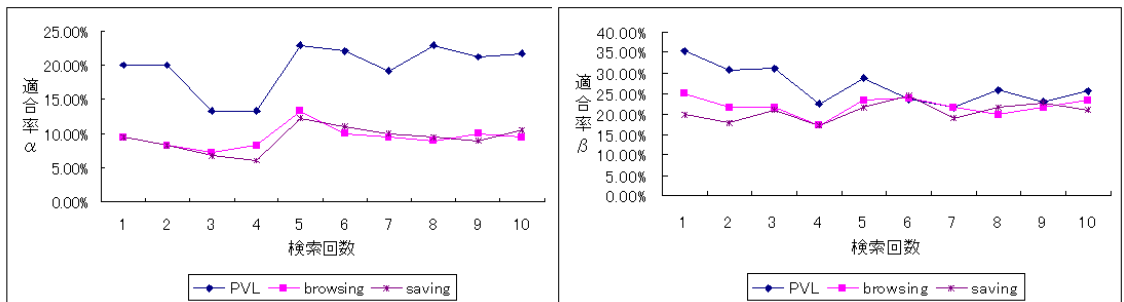


図 4 ブラウジング操作のみ、保存操作のみをフィードバック利用した際の適合率 α, β の推移
 Fig.4 Movement of precision factors α, β of saving and browsing (feedback only
 browsing: left, feedback only saving: right).

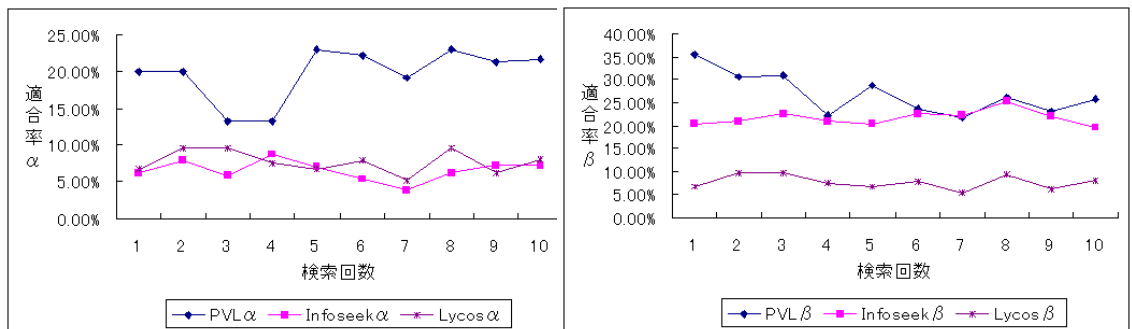


図 5 適合率 α, β の推移
 Fig.5 Movement of precision factors α, β of saving and browsing.

実験では、上位 30 件のデータを対象に要・不要の判断を行った。したがって、上式の分母(検索結果の数)を 30 として計算を行う^{3),4)}。

実験では、あらかじめカテゴリを“computer”に設定し、キーワードは自由に入力してもらった。ただし、PVL 利用時と検索エンジン(インフォseek, Lycos Japan) 利用時では同じキーワードを用いて繰り返し検索を行った。

検索結果の中から、利用者独自の判断で、要・不要を判断し、表示する実データの決定と、本欄に保存するデータの決定を行った。

5.2.2 実験結果

本実験により得た結果を示す。4.2 節で述べた得点付けのための利用者による対話的操作(操作 1)(操作 2)の、検索結果の精度向上に対する貢献度を検討するため(操作 1)のみによる得点付けを行いフィードバック利用した場合と(操作 2)のみによる得点付けを行いフィードバック利用した場合の実験を行い、それぞれ適合率 α, β の追跡を行った。

図 4 左に 10 人の利用者の格納に関する適合率 α の平均値の推移を示す。また、図 4 右では、表示に関する適合率 β の推移を示す。は(操作 1)(操作 2)

ともに得点付けを行いフィードバックしたとき、は(操作 2), *が(操作 1)のみによる得点付けを行いフィードバック利用した場合の適合率 α, β の推移を表す。

この結果により(操作 1)(操作 2)は、ほぼ同じ値を推移するため、それぞれのシステムへの貢献度は同等であると見なし、以下では(操作 1)(操作 2)に同じ値(1 点)を用いて実験を行った。

図 5 左に 10 人の利用者の格納に関する適合率 α の平均値の推移を示す。また、図 5 右では、表示に関する適合率 β の推移を示す。図 5 において、縦軸は適合率 α, β を示し、横軸は検索回数を示す。また、は本システム利用時、*が各々検索エンジン(インフォseek, Lycos Japan) に対応するエージェントのみを利用したときの適合率 α, β の推移を表す。インフォseek, Lycos Japan の実験は、それぞれ 1 つのエージェントだけを動作させて実験を行った。

この結果より、本システムは既存の検索エンジンと比較して適合率 α, β ともに増加している。また、表 2 に、既存の検索エンジンと、本システムの適合率 α, β の差を利用回数ごとに示す。それぞれの差を明らかにするため、あらかじめ既存の検索エンジンの 1 つで

表 2 既存の検索エンジンとの適合率の差

Table 2 Comparison of the precision factors α , β of existing search engine and PVL.

		1 回目	2 回目	3 回目	4 回目	5 回目	6 回目	7 回目	8 回目	9 回目	10 回目
PVL	適合率 α	1.95	1.60	1.01	1.02	1.55	1.30	1.53	0.91	1.51	1.03
	適合率 β	1.36	1.17	1.04	1.00	1.36	1.07	0.96	1.04	1.05	1.23
Infoseek	適合率 α	0.94	0.82	0.60	1.17	1.06	0.68	0.73	0.65	1.15	0.88
	適合率 β	1.02	1.10	1.15	1.12	1.10	1.11	0.79	0.85	0.91	1.12
Lycos	適合率 α	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	適合率 β	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

表 3 同一キーワードでの 1 回目と 11 回目の適合率 α , β の変化 (computer カテゴリ)Table 3 1-st and 11-th precision factors α , β for same keywords (category "computer").

	1 回目	11 回目	変化率
適合率 α (保存)	11.67%	15.00%	約 28.6%増加
適合率 β (表示)	30.56%	29.44%	約 4.6%減少

表 4 同一キーワードでの 1 回目と 11 回目の適合率 α , β の変化 (sports カテゴリ)Table 4 1-st and 11-th precision factors α , β for same keywords (category "sports").

	1 回目	11 回目	変化率
適合率 α (保存)	5.00%	5.83%	約 16.7%増加
適合率 β (表示)	20.00%	19.17%	約 4.2%減少

ある Lycos の適合率 α , β を 1 とし、それぞれの比率を計算した。しかし、一部前の操作と比較して適合率の減少が見られ、利用を進めるに従ってつねに適合率が上昇しているといい難い。これらは検索キーワードにより、適合率の増減があると考えられる。

そこで、フィードバック効果を確認するため、各利用者が 10 回の検索終了後に、再び 1 回目の検索時に用いたキーワードによって検索を行った際の、適合率 α , β の変化率を表 3, 表 4 に示す。

図 5 では、適合率の増減は既存の検索エンジンを利用する場合と同様、キーワードによって適合率の増減が見られた。そこで、本実験を行うことにより、キーワードに依存した適合率の増減をなくし、本手法の有効性を確認した。本実験により、表示に関する適合率 β は約 4.6%減少したが、格納に関する適合率 α は約 28.6%増加することを確認した(表 3)。

また、異なるカテゴリでの効果を確認するため、sports カテゴリにおいて上記と同様の実験を行った結果、適合率 α は約 16.7%増加、適合率 β は約 4.2%減少となった(表 4)。

双方とも適合率 β に関する数値は減少しているが、computer カテゴリの結果を用いて、

$$\text{適合率 } \gamma = \frac{\text{本欄に格納した数 } (C)}{\text{実データを表示した数 } (B)} \quad (5)$$

によって適合率 γ の計算を行うと、1 回目の適合率 γ は 40.66%、11 回目は 47.95%となり約 17%増加することを確認した。

表 3, 表 4 において適合率 β が減少しているのは、被験者が過去の試行においてブラウズした実データを記憶しており、不適合と判断したデータを、以後の試行では表示しなかったことが原因であると推測される。

したがって、提供された 30 件の中で実データを表示する数は減少するが、個別化が進んだ状態のため、表示した情報の中に利用者の要求に見合う情報が多く含まれることが考えられる。

6. 関連研究との比較

個別化を目指す様々な取り組みが行われている¹⁾。清光らは文献 2) において、WEB データを利用者に応じて個別化し再構成する手法と、利用者の WEB データをブラウズする環境に依って再構成する手法に着目し、個別化に関するアプローチの分類を行っている。PVL はクライアント側で個別化する特徴を持つため、まず、既存の個別化機構を持つシステムをサーバ側で個別化を行う例と、クライアント側で行う例に分類し、PVL と比較を行う。

クライアント側での個別化を実現するための取り組み例として文献 13)~16) などがある。金井ら¹³⁾ の、ベクトル空間モデルを利用し、個人書棚の作成を目的とした研究があげられる。この取り組みでは、3D インタフェースを提供され、ブラウジング検索を行っている。また、Jayawardana ら¹⁴⁾ の取り組みは、デジタル図書館を対象にし、注釈付けや編集、要約、リンク付けなどを可能とし、PDM モデル (Personal Document Model) を提案している。また、斥候型エージェントとして Letizia と PowerScout がある¹⁵⁾。Letizia はクライアントの操作履歴を用いる zero-input ソフトウェアで、利用者の閲覧行動から好みをリアルタイムに解析し、利用者の好みにあった情報を推薦するシステムである。Kerschberg らは、意味的分類を用いた、個別化可能なメタ検索エンジン WebSifter II について述べている¹⁶⁾。

一方、サーバでの個別化を実現するための代表的な取り組みでは、神場ら¹⁷⁾の操作履歴と記事スコアを用いて、コンテンツの加工による個別化を目的とした取り組みがある。

ベクトルの類似度計算によって個別化情報提供を可能にする InfoBroker¹⁸⁾では、TF/IDF や概念の軸を用いて、ニュース記事の個別化を行っている。Takealook¹⁹⁾では、興味空間の可視化によって情報探索の支援を行っている。この取り組みでは、ユーザとの対話によって提供の個別化を実現している。さらに、視点と特徴ベクトルを利用して、文書整理を個別化する FishView²⁰⁾などの取り組みがある。

これらと比較して PVL では本棚メタファを用いて、インターネット上の情報資源を個別化して自分専用の本棚を構築する。利用者の対話的操作によって付けられた、各アクセスエージェントの信頼度は得点として利用者プロフィールに記述され、並べ替えや、比例配分を行い、システムに組み込みクライアント側での個別化を実現している。

また、情報検索の分野において、利用者の検索を支援するために、過去の試行から利用者の好みや目的を抽出する適合フィードバックに関する研究が多く行われている²¹⁾。江口らは Rocchio の式を拡張し、適合フィードバックのパラメータを動的に調整し、利用者の検索目標が多様かつ動的に変化する場合や、漠然とした検索目標の場合にも、WWW 情報検索に適合フィードバックが有効であることを確認している²²⁾。ここでは、利用者の対話的操作から、興味を検出し暗黙的にフィードバック利用している。一方、仲川らは利用者が個人の目的に合わせてランキングを調整する手法を提案している。これは、提供された文書リストに対し、利用者が直接ランキングの調整操作を行い、スコア計算に反映させる。検索結果の適合率と再現率の重み付き調和平均から求められる E 尺度の概念を用いて、ユーザの調整意図を反映した新しい質問ベクトルの生成を行っている²³⁾。

これらの手法に対し、PVL では利用者の表示・格納といった対話的操作によって、アクセスエージェントに対する得点付けを行い、暗黙的にフィードバックを行っている。利用者は個別化するために特別な操作が必要ではなく、利用者負担のかからない枠組みである。また、個別化機構を持たない情報資源に対して適用することが可能である。

メタ検索エンジンの実システムとして代表的な SavvySearch²⁴⁾では、検索キーワードに対する検索エンジンのランク付け、ネットワークの負荷や検索エ

ンジンの状況などを用いて、約 13 個の検索エンジンの中から、3 つの適切な検索エンジンの組合せを動的に決定する。また、MetaCrawler²⁵⁾は、7 個の検索エンジンで並列に検索を行い、得点を正規化し、スコア順に利用者に提供する。PVL では、固定した 4 つの検索エンジンを利用している。各検索エンジンに対するアクセスエージェントを備えており、カテゴリごとにアクセスエージェントの得点を算出し、個別化した情報を提供する。

7. おわりに

本稿では、本棚メタファを用いてインターネット上の情報を利用するプライベート仮想ライブラリ PVL を提案した。

PVL は、利用者の対話的操作によってアクセスエージェントに対する得点付けを行うため、利用者は本システムを繰り返し利用することで、意識することなく個別化を進める。

また、プロトタイプシステムを構築し、格納に関する適合率と表示に関する適合率を用いて評価実験を行った。実験結果より、既存の検索エンジンと比較して、10 回の検索時の平均値を用いると、格納に関する適合率が約 34.1%、表示に関する適合率が約 12.8% 上昇することを確認した。さらに、適合率は検索時に用いるキーワードによって増減する可能性を考え、10 回検索を行う実験が終了した後、さらに 1 回目に用いた検索キーワードによって検索を行った。その結果、表示に関する適合率は約 4.6% 減少したが、格納に関する適合率は約 28.6% 増加した。

今後の課題として、本棚の見せ方の工夫、より進んだ本棚の管理法、3D 形式の GUI への情報の配置手法などがあげられる。

謝辞 日頃ご指導いただく矢島脩三教授に深甚なる謝意を表す。また、実験に協力いただいた上島研究室諸氏に謝意を表す。本研究の一部は、2002 年度関西大学重点領域研究助成金、文部科学省科学研究費補助金(課題番号 14023115)による。

参考文献

- 1) Special Issue on Personalization, *Comm. ACM*, Vol.43, No.8 (2000).
- 2) 清光英成, 竹内淳記: Web データの個別化と環境適応, 情報処理学会論文誌: データベース, Vol.42, No.SIG 8 (TOD 10), pp.185-194 (2001).
- 3) Ueda, M. and Ueshima, S.: Client-side Personalization Based on Agent Scoring in Private

- Virtual Library PVL, *4th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT2001)*, pp.31-35 (2001).
- 4) Ueda, M., et al.: Bottom-up Generation of Private Virtual Library PVL over Advanced Information Network, *INFORMS & KORMS Seoul 2000 (Korea), CD Proceedings* (2000).
 - 5) Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, Addison-Wesley (1999).
 - 6) 守村 篤, 清木 康: WWWを対象としたサーチエンジン間統合機能の実現, 夏のデータベースワークショップ 2001 (DBWS2001), 125-15 (2001).
 - 7) 九津見洋, 内藤榮一, 荒木昭一, 江村里志, 新居薫治: ユーザ適応型ホームページ推薦ソフト“ウェブナビゲーター”の開発, 電子情報通信学会論文誌 D-II, Vol.J84-D-II, No.6, システム開発論文特集, pp.1149-1157 (2001).
 - 8) Yahoo! Japan. <http://www.yahoo.co.jp>
 - 9) Ueda, M., Yoshida, T. and Ueshima, S.: Browsing Interface of Private Virtual Library PVL, 情報処理学会第 63 回全国大会論文集 (3), pp.9-10 (2001).
 - 10) インフォシーク. <http://www.infoseek.co.jp>
 - 11) エキサイト. <http://www.excite.co.jp>
 - 12) Lycos Japan. <http://www.lycos.co.jp>
 - 13) 金井秀明, 箱崎勝也, 石川克則: 利用者特性とドキュメント特性を利用したブラウジング環境, 情報処理学会論文誌: データベース, Vol.41, No.SIG3 (TOD6), pp.46-57 (2000).
 - 14) Jayawardana, C. and Hirakawa, M.: Interface Mechanism to Personalize a Digital Library, 情報処理学会論文誌, Vol.41, No.10, pp.2863-2872 (2000).
 - 15) Lieberman, H., Fry, C. and Weitzman, L.: Exploring the Web with RECONNAISSANCE AGENTS, *Comm. ACM*, Vol.44, No.8, pp.69-75 (2001).
 - 16) Kerschberg, L., Kim, W. and Scime, A.: A Semantic Taxonomy-Based Personalizable Meta-Search Agent, *2nd International Conference on Web Information Systems Engineering (WISE2001)* (2001).
 - 17) 神場知成, 小池雄一, 古閑義幸: 情報のパーソナライゼーションとその記述方式, 人工知能学会誌, Vol.41, No.6, pp.936-942 (1999).
 - 18) 橋高博行, 佐藤直之, 鈴木英明, 曾根岡昭直: パーソナライズ情報提供方式の提案と評価, 情報処理学会論文誌, Vol.40, No.1, pp.175-187 (1999).
 - 19) 角 薫, 角 康之, 間瀬健二, 中須賀真一: 個人の概念空間を利用した興味と推定による情報提供, 電子情報通信学会論文誌 D-II, Vol.J82-D-II, No.10, pp.1634-1644 (1999).
 - 20) 高間康史, 石塚 満: FISH VIEW システム: 概念体系に基づく視点情報を活用した文書整理支援, 情報処理学会論文誌, Vol.41, No.7, pp.1976-1986 (2000).
 - 21) Rocchio, J.J.: Relevance feedback in information retrieval, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Salton, G. (Ed.), pp.313-323, Prentice Hall (1971).
 - 22) 江口浩二, 伊藤秀隆, 隈元 昭: ユーザへの適応性を考慮した適合フィードバックによる WWW 情報検索, 電気学会論文誌 (C), Vol.117-C, No.11, pp.1643-1649 (1997).
 - 23) 仲川こころ, 木下敦史, 高田義朗, 関 浩之: ユーザによる文書ランキングの調整が可能な対話的 WWW 検索支援手法の提案, 情報処理学会第 63 回全国大会論文集 (3), pp.131-132 (2001).
 - 24) Howe, A.E. and Dreilinger, D.: SAVVY-SEARCH: A Metasearch Engine That Learns Which Search Engines to Query, *AI Magazine*, Vol.18, No.2, pp.19-25 (1997).
 - 25) Selberg, E. and Etzioni, O.: Multi-Service Search and Comparison Using the MetaCrawler, *4th Int. WWW Conference*, P05-2 (1995).

(平成 14 年 6 月 21 日受付)

(平成 14 年 10 月 11 日採録)

(担当編集委員 清木 康)



上田真由美 (学生会員)

1974 年生. 1998 年関西大学総合情報学部卒業, 2000 年同大学大学院総合情報学研究所修士課程修了. 現在, 同博士後期課程在学中. Web 情報システム, 個別化技術等に興味を持つ. ACM 学生会員.



上島 紳一 (正会員)

1955 年生. 1978 年京都大学工学部数理工学科卒業, 1983 年同大学院工学研究科博士課程単位取得退学. 1994 年関西大学総合情報学部教授, 現在に至る. 1998 年から 1999 年米国ジョージメイソン大学客員研究員. 1985 年日本自動制御協会榎木記念賞論文賞. 工学博士. システムモデリング, 情報ベース等に興味を持つ. 電子情報通信学会, システム制御情報学会, ACM, IEEE Computer Society 等の会員.