

平成 12 年度

学士学位論文

# 個人適応型情報提供支援システム

Personalized Information Navigator

1010446 間城 昌厚

指導教員 清水 明宏

2001 年 2 月 5 日

高知工科大学 情報システム工学科

## 要 旨

### 個人適応型情報提供支援システム

間城 昌厚

インターネットの普及に伴い，ネットワーク上には大量の情報が蓄積されている．そのため，ユーザが必要とする情報を探し出すことが困難になりつつある．現在，ユーザの求める情報を簡易に探し出すための技術が開発されている．しかし，検索結果の信用性やコンテンツへの到達時間，利用環境の制約などの問題点が解決されていない．従って，環境に依存することなく，利用者が必要とする情報を適切に抽出し，提供する技術の開発が望まれている．

本稿では，多種多様な環境下におけるユーザに対して満足のゆく最適なプロフィール項目を抽出しとフィルタリング技術を確立することを目的とする．

まず，プロフィール項目をリアルタイムに変化しない住所や趣味，情報内容などを静的プロフィールとし，どのコンテンツが選択されるかというアクセス履歴など，確率的に変動する情報を動的プロフィールとした．そして，フィルタリングを行う際の基盤となる実験環境の構築を行った．

キーワード フィルタリング，静的プロフィール，動的プロフィール

# Abstract

## Personalized Information Navigator

Masaatsu MASHIRO

As the Internet is popularized, a large quantity of information goes onto the network. Therefore, it can't get the information which a user is necessary for. The information which a user looks for is looked for simply at present technology. But, there are reliability of the reference result, arrival time to the information, a problem such as the restriction of the use environment. Therefore, the way of solving it toward them is being looked for.

In this paper The professional most suitable file item which it is satisfied with toward the user in the bottom of the various environments is extracted, and it aims at establishing filter ring technology .

First, a professional file is divided into the professional dynamic file with the professional file whose it is static. Then, building of the experiment environment which becomes a base to do a filter ring was done.

**key words** Filtering,Static Profile,Dynamic Profile

# 目次

第 1 章	はじめに	1
1.1	情報収集の現状 . . . . .	1
1.2	情報検索システム . . . . .	1
1.3	情報フィルタリング . . . . .	2
1.4	協調フィルタリング . . . . .	3
1.5	ブックマークエージェント [2] . . . . .	4
第 2 章	研究内容	6
2.1	研究内容 . . . . .	6
2.2	研究項目 . . . . .	6
第 3 章	プロファイルの初期設定	7
3.1	ユーザプロファイル . . . . .	9
3.1.1	静的ユーザプロファイル . . . . .	9
3.1.2	動的ユーザプロファイル . . . . .	10
3.2	コンテンツプロファイル . . . . .	11
3.2.1	静的コンテンツプロファイル . . . . .	11
3.2.2	動的コンテンツプロファイル . . . . .	12
第 4 章	静的フィルタリング	13
4.1	フィルタリング . . . . .	13
第 5 章	実験環境の構築	16
5.1	実験環境における機能概要 . . . . .	16
5.2	実験環境の構築 . . . . .	19

## 目次

5.2.1	Postgresql . . . . .	19
5.2.2	Java2 . . . . .	19
5.2.3	JDBC . . . . .	20
<b>第 6 章</b>	<b>静的プロファイル項目への登録</b>	<b>21</b>
6.1	静的プロファイル項目の登録 . . . . .	21
6.1.1	静的ユーザプロファイル . . . . .	22
6.1.2	静的コンテンツプロファイル . . . . .	23
<b>第 7 章</b>	<b>まとめと今後の展望</b>	<b>24</b>
7.1	動的プロファイルの蓄積, 更新方法 . . . . .	24
7.2	初期設定プロファイルの検討 . . . . .	25
7.3	フィルタリングアルゴリズムの改良 . . . . .	25
7.4	付加機能 . . . . .	26
<b>謝辞</b>		<b>27</b>
<b>参考文献</b>		<b>28</b>
<b>付録 A</b>		<b>30</b>
A.1	静的フィルタリングプログラムソース . . . . .	30

# 目次

1.1	情報検索システム	2
1.2	情報フィルタリングモデル	3
1.3	協調フィルタリングモデル	4
1.4	ブックマークエージェントのシステム構成	5
4.1	プログラム構成	14
5.1	実験環境	18
5.2	JDBC	20
6.1	静的ユーザプロフィール登録処理の流れ	22
6.2	静的コンテンツプロフィール登録処理の流れ	23
7.1	動的プロフィールの蓄積例	25

# 表目次

3.1	静的ユーザプロフィール . . . . .	9
3.2	動的ユーザプロフィール . . . . .	10
3.3	静的コンテンツプロフィール . . . . .	11
3.4	動的コンテンツプロフィール . . . . .	12

# 第 1 章

## はじめに

### 1.1 情報収集の現状

近年，インターネットの急激な普及にともない World Wide Web(WWW) 上には，様々な情報が流出している．したがって，ブラウジングのみで，各ユーザが求めるコンテンツにたどり着くことは非常に困難である．現在では，このような問題を解決する様々な情報収集方法が開発されている．本章では，従来の情報収集方法をふまえたうえ，今後，要求される技術について考察する．

### 1.2 情報検索システム

情報検索とは，検索式の修正と検索結果に対するフィードバックの繰り返しによりユーザの求めるデータを特定していく情報収集法である．ユーザは求めるデータを表す検索式を立て，データベースに問い合わせる．データベースは検索式に合うデータをユーザに返戻する．ユーザは送られてきたデータ（検索結果）を評価する．ユーザの要求を満たす場合，ユーザは検索を終了する．検索結果がユーザの要求を満たさない場合，検索式を修正したり，検索結果の中から要求に近いデータを特定し（フィードバック），再びデータベースに問い合わせる．情報検索は以上の繰り返しによりユーザの求めるデータを特定していく情報収集法である．

情報検索において，個々の検索過程及びユーザは独立である．このため，同じ関心を持つユーザは互いに情報交換あるいは情報共有できない．効率的な検索は検索者個人の知識と

### 1.3 情報フィルタリング

熟練に委ねられている。

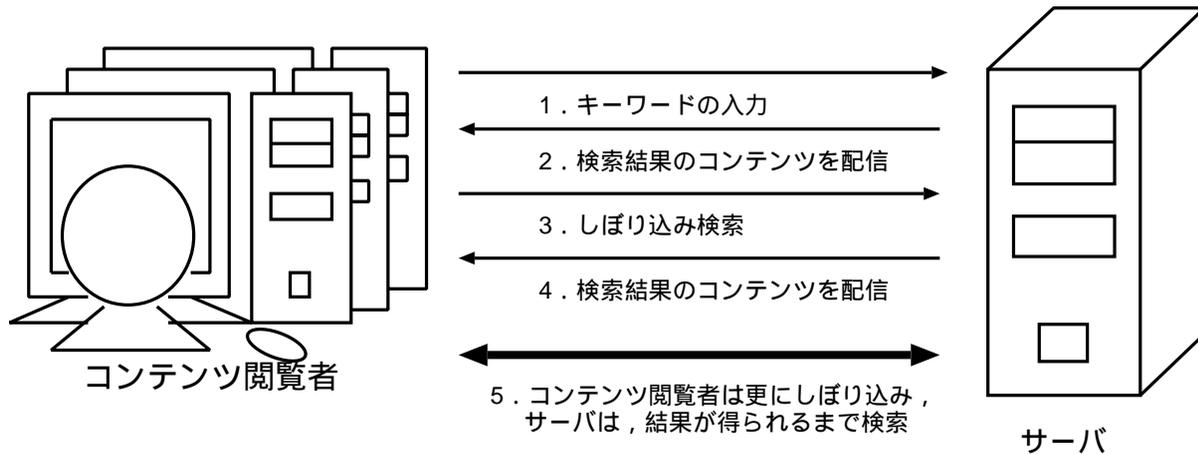


図 1.1 情報検索システム

### 1.3 情報フィルタリング

情報フィルタリングは、大量の情報の中から利用者が必要とする情報を、システムが自動的に、または必要に応じて利用者との対話を用いて取舍選択する技術である。情報フィルタリングの概要は次のとおりである。

- 利用者は必要に応じて、要求する情報に関する趣向をプロフィールとして事前に記述しておく。
- システムは、利用者に対話的にキーワードなどを含む検索式を入力してもらうことにより、プロフィールと検索式に適合するデータを利用者に提供する。

## 1.4 協調フィルタリング

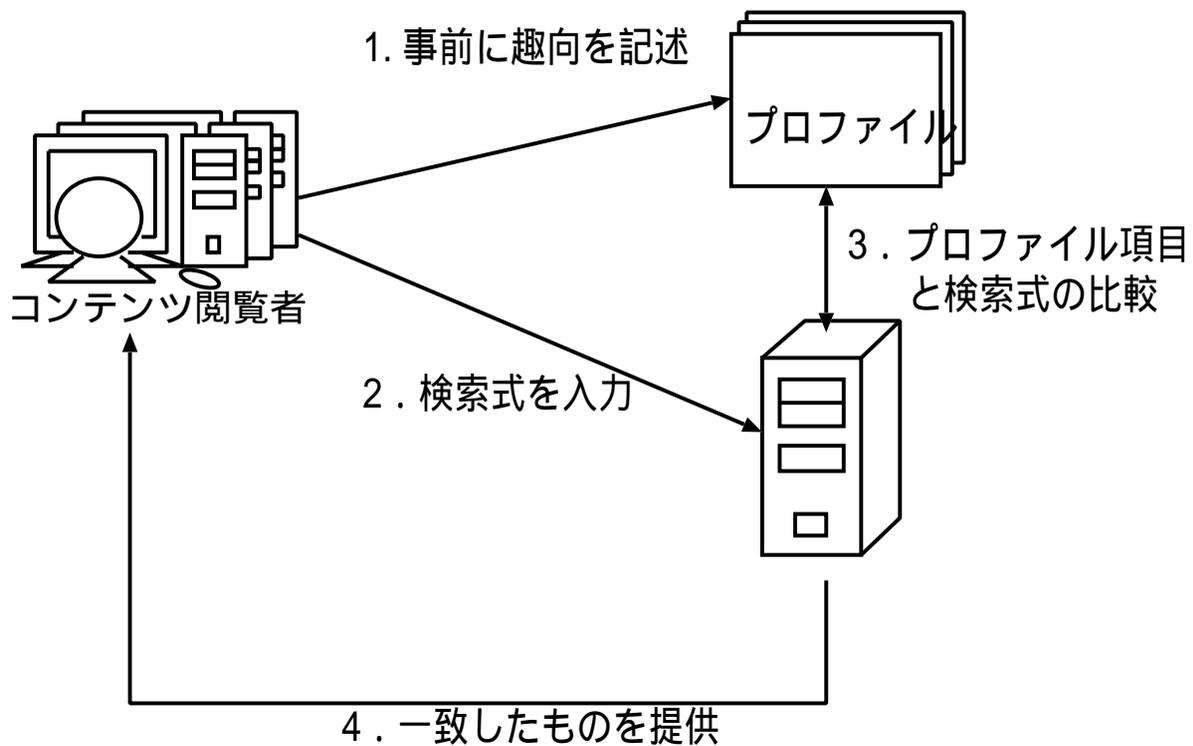


図 1.2 情報フィルタリングモデル

## 1.4 協調フィルタリング

協調フィルタリングは、推薦システムとも呼ばれ、文献の内容には関わらず、文献に対して他者から与えられた評価に着目した手法である。典型的には、各利用者に対し、似た興味を持つ他の利用者から与えられた評価を得ている文献の推薦が、フィルタリングと見なされている。協調フィルタリングは、コミュニティ内における利用者間の協調活動を利用しており、言い替えると他者の知識を用いた情報収集手法である。

協調フィルタリングは、他の利用者に対する評価に基づくため、既に高い評価を得ている文献の入手には適している。また、索引づけが不要であるため、索引語の付与が特に困難な文献（画像や映像）の収集には、内容に基づくフィルタリングに対して圧倒的に優る。反面、他の利用者の評価を待たなくてはならないので、文献がフィルタリング可能になるまでの時間がかかる。更に、文献の意味内容は保証されない。

## 1.5 ブックマークエージェント [2]

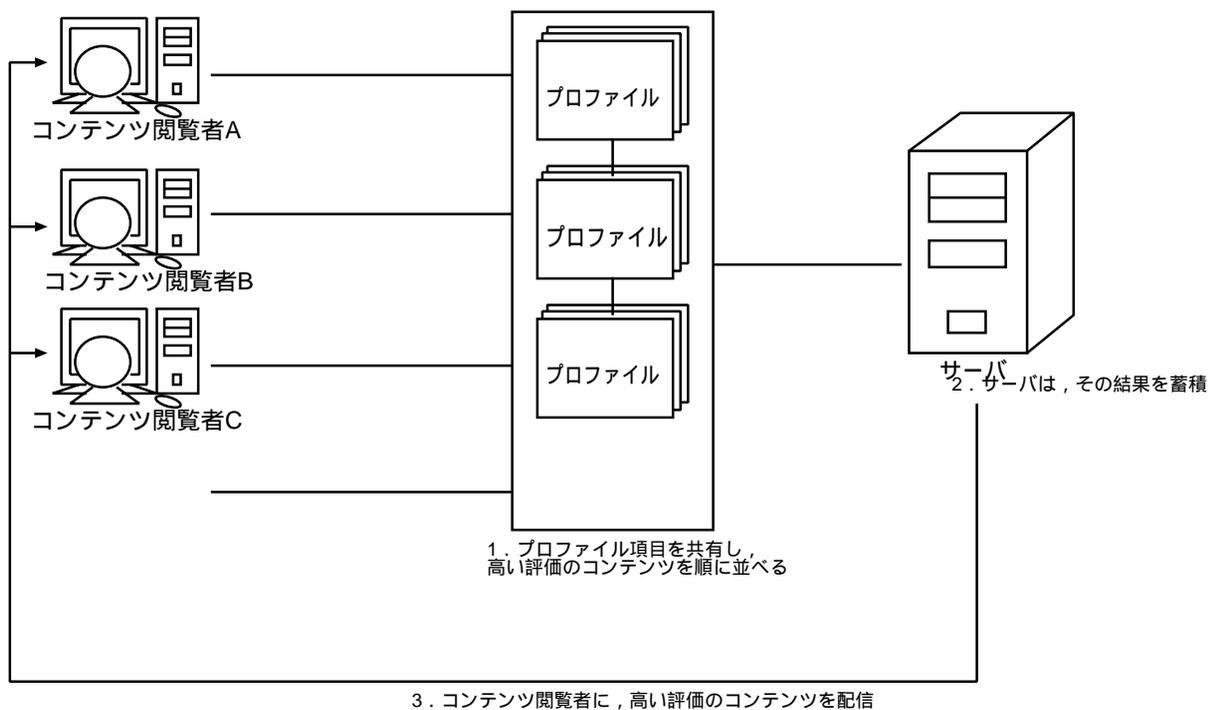


図 1.3 協調フィルタリングモデル

## 1.5 ブックマークエージェント [2]

ブックマークエージェントは、ユーザの明示的要求を必要とせずに、ユーザの欲する URL 情報を提示できる。また、ユーザは、ブックマークエージェントに、明示的にキーワードを提示することも可能である。ここで、留意して欲しいことは、ブックマークエージェントが類似ページを検索する範囲は、興味がある程度共通していると想定される小規模なグループのユーザのブック・マークに限定されていることである。このことは、あるキーワードによる検索において、共通の興味という文脈を利用した、絞り込みが行われていることを意味し、一般の検索エンジンよりも効果的な情報フィルタリングが期待できる。また、この分野の先行研究では、ユーザが次にたどるであろうページに対する予測の学習に主眼が置かれているが、この研究では、グループ間の情報共有に主眼を置いている。

グループ間で情報を共有するとともに、情報検索を行うシステムで、特定のグループ間では、URL の共有ができるので効率がよい。しかし、不特定のグループ間では、興味の領域が一致しないので、汎用性に乏しい。

## 1.5 ブックマークエージェント [2]

ブックマークエージェントのシステム構成図は、1.4 に示す。

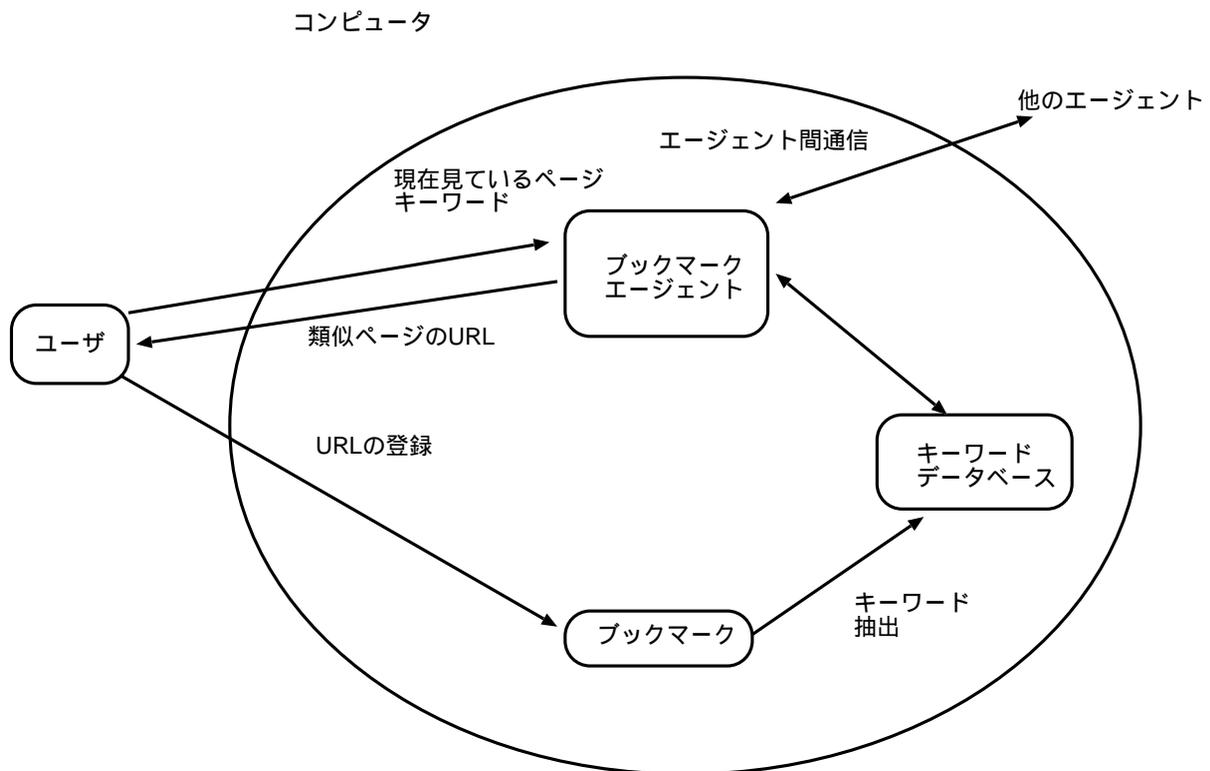


図 1.4 ブックマークエージェントのシステム構成

以上のように、従来の技術では、検索結果の信用性や、コンテンツへの到達時間、利用環境の制約などの問題点が解決されないままである。よって、環境に依存することなく、利用者が必要とする情報を適切に抽出し、提供する技術の開発が望まれている。

## 第 2 章

# 研究内容

### 2.1 研究内容

本研究では、従来技術の問題点を解決するためにでは成し得なかった、多種多様なユーザに対して満足のゆくフィルタリング技術を確立することを目的とし、その基盤となる実験環境の構築を行うことにする。

### 2.2 研究項目

1. プロフィルの初期設定
2. 実験環境の構築
3. 静的フィルタリングの作成

## 第 3 章

# プロフィールの初期設定

多種多様なユーザとコンテンツを的確に結ぶ方式を開発するにあたり、フィルタリング処理のアルゴリズムを決定する前に、個人が持つプロフィールとコンテンツが持つプロフィールにどういったものを用意しなければならないかを考察した。

利用者の立場になりユーザプロフィールの作成を進めなければならない。よって、ユーザの一般的な生活状態について調べる必要がある。今回は、総務省 統計局 統計センターの調査を利用することにした。

総務省の調べによると国民のライフスタイルは、年齢、性別、休曜日などによってそれぞれのライフスタイルが異なるがわかった。このことにより、インターネット上でも、各々のライフスタイルにあわせたコンテンツ要求が考えられる。そこで、ユーザプロフィールでは、統計データを参考にして、ユーザ ID の他に、年齢、性別、出身地、現住所、趣味、得意分野、休曜日、家族構成に分けることとした。

次に、各ユーザの求めるコンテンツプロフィールについて考察した。コンテンツプロフィールには、先程の、ユーザプロフィールに対応するプロフィールを作成しなければならない。今回、初期設定コンテンツプロフィールには、コンテンツ ID、コンテンツ登録者の ID、コンテンツカテゴリ、コンテンツカテゴリ ID、タイトル、コンテンツの対象年齢、コンテンツの評価、情報内容、ホームページアドレス、予想検索キーワード、コンテンツ内容の関連地域を設けることとした。

しかし、今後このプロフィールを使用して、フィルタリングを続けてゆくととなると単純な年齢層、性別などによる統計的なフィルタリング結果になることは明らかである。このことは、各利用者にあわせたシステムの構築がなされない、という問題点ができる。

故に、今回、利用者がどのコンテンツを選択するかというアクセス履歴など、確率的に変動する情報を動的プロフィールとして、一人一人のユーザに合わせたフィルタリングを行うことを考案した。

動的プロフィールの初期設定は、ユーザ側が、ユーザ自身のコンテンツアクセス履歴、コンテンツ側は、コンテンツに対する様々なユーザコンテンツアクセス履歴とユーザからのコンテンツ評価とした。

それぞれの静的プロフィールと動的プロフィールについて表を 3.1, 3.2, 3.3, 3.4, 示す。

### 3.1 ユーザプロフィール

## 3.1 ユーザプロフィール

### 3.1.1 静的ユーザプロフィール

表 3.1 静的ユーザプロフィール

プロフィール項目	利用方法
ユーザ ID...例 (6桁の数値)	ユーザ確認と DB の検索に使用
年齢 (生年月日 20010101)... 8桁の数値	ユーザのコンテンツ利用に対する参考, 今後のフィルタリングに参考
性別...例 (男 = 1, 女 = 2)	同上
出身地...郵便番号 (7桁)	動的コンテンツプロフィールのコンテンツ内容の関連地域とフィルタリング
現住所...郵便番号 (7桁)	同上
趣味 1... 例 (スポーツ)	選択方式で, 静的コンテンツプロフィールの情報内容とのフィルタリング
不得意分野... 例 (経済)	選択方式で, 静的コンテンツプロフィールの情報内容とのフィルタリング
休曜日... 例 (曜日を選択)	選択内容を動的プロフィールの更新とともに使用
家族構成... 例 (子どもが何人 いるかなどを選択)	動的プロフィールの更新とともに使用

### 3.1 ユーザプロフィール

#### 3.1.2 動的ユーザプロフィール

表 3.2 動的ユーザプロフィール

プロフィール項目	利用方法
ユーザ ID (静的ユーザプロフィールのユーザ ID と同じ内容)	ユーザ ID の確認と DB の検索に使用
コンテンツアクセス履歴	各ユーザのコンテンツアクセス履歴を用いてコンテンツの人気度をはかる

### 3.2 コンテンツプロフィール

## 3.2 コンテンツプロフィール

### 3.2.1 静的コンテンツプロフィール

表 3.3 静的コンテンツプロフィール

プロフィール項目	利用方法
コンテンツ ID... 例 (11 桁の 数値)	コンテンツの確認と DB の検索に使用
コンテンツ作成者の ID	動的プロフィールのコンテンツ評価ポイントとフィルタリング を行いランキング機能に使用する
タイトル	静的ユーザプロフィールの趣味とでフィルタリング
情報内容	静的ユーザプロフィールの趣味とでフィルタリング
ホームページのアドレス	コンテンツ閲覧の際に表示，作成者名との照合
予想検索キーワード	検索機能の際にユーザからのキーワードと照合
コンテンツ内容の関連地域	静的ユーザプロフィールの出身地，現住所とでフィルタリン グ

## 3.2 コンテンツプロフィール

### 3.2.2 動的コンテンツプロフィール

表 3.4 動的コンテンツプロフィール

プロフィール項目	利用方法
コンテンツ ID(静的コンテンツプロフィールのコンテンツ ID と同じ内容)	コンテンツの確認と DB 検索に使用
コンテンツアクセス履歴	コンテンツのアクセス履歴を用いてコンテンツの利用度の多い順に配信するようににする
コンテンツ評価 (この項目は、コンテンツの閲覧の際に別ウィンドで評価されている形にしたい)	コンテンツ利用の最後にコンテンツの評価を入力してもらいコンテンツの人気度の順に配信する

## 第 4 章

# 静的フィルタリング

### 4.1 フィルタリング

今回考案する．フィルタリング機能は，以下の通りである．

1. ユーザが，ユーザ ID を入力するとユーザプロフィール中のユーザ ID と一致する項目の趣味 ID を取り出す．

次に，趣味 ID をより，コンテンツカテゴリのコンテンツカテゴリ ID を検索する．

最後に，コンテンツテーブルと一致する，「コンテンツ項目」をユーザに返す．

結果の表示は，以下の通りである．

- ユーザ ID
- ユーザ名
- コンテンツタイトル
- コンテンツ情報内容
- ホームページアドレス

処理の流れ，フィルタリング結果のコンソール上で表したものについては，次ページ 4.1 に記載する．

## 4.1 フィルタリング

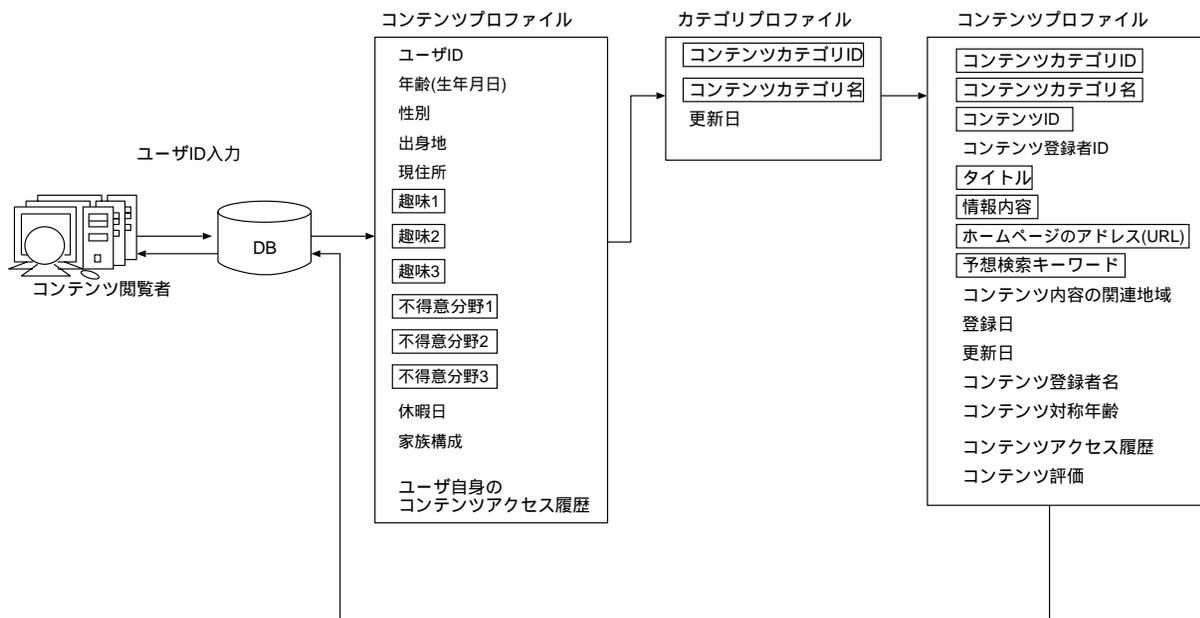


図 4.1 プログラム構成

## 4.1 フィルタリング

フィルタリングされた結果をコンソール上で表すと以下のようなになる。

```
[mashiro@lupus jdbc]$ java filtering jdbc:postgresql://localhost:5432/kojin postgres ""
```

学籍番号を入力してください。

1010446

趣味：音楽

タイトル	情報内容	アドレス
トラットリア	渋谷系音楽	<a href="http://www2.eccosys.co.jp/~kawakazu/new/index.html">http://www2.eccosys.co.jp/~kawakazu/new/index.html</a>
ギョガンレンズ	ガレージ	
54NUDEHO		

# 第5章

## 実験環境の構築

### 5.1 実験環境における機能概要

初期段階の実験環境を構築するにあたり、ユーザ管理機能、コンテンツ管理機能、フィルタリング機能、コンテンツ検索機能、ランキング機能、要求受付機能、画面表示機能の機能、DB アクセス機能に分けることにした。

- ユーザ管理機能である  
各コンテンツ閲覧者と各コンテンツ登録者のシステムに対する情報入力機能である。
- コンテンツ管理機能  
コンテンツの登録、情報取得や情報削除の機能ができる。
- フィルタリング機能  
ユーザ管理機能とコンテンツ管理機能に蓄積されている情報内容を基に、コンテンツ閲覧者とコンテンツ登録者への情報提供を行う機関である。
- コンテンツ検索機能  
通常の検索キーからのコンテンツ取得のできる機能である。
- ランキング機能  
各ユーザのコンテンツへのアクセス履歴を基に、コンテンツカテゴリごとに分けアクセス頻度の高い順に並び変える。
- 要求受付機能  
コンテンツ検索、ランキング表示要求の受付を行う機関である。

## 5.1 実験環境における機能概要

- 画面表示機能

フィルタリングされたコンテンツの一覧を表示，検索画面，検索結果，ランキング表示を各ユーザに表示する機能である．

- DB アクセス機能

ユーザ管理，コンテンツ管理，フィルタリング，コンテンツ検索，ランキングのそれぞれのデータを蓄積，更新をする機能である．

それぞれの機能の構成図は，図 実験環境??に示す．

## 5.1 実験環境における機能概要

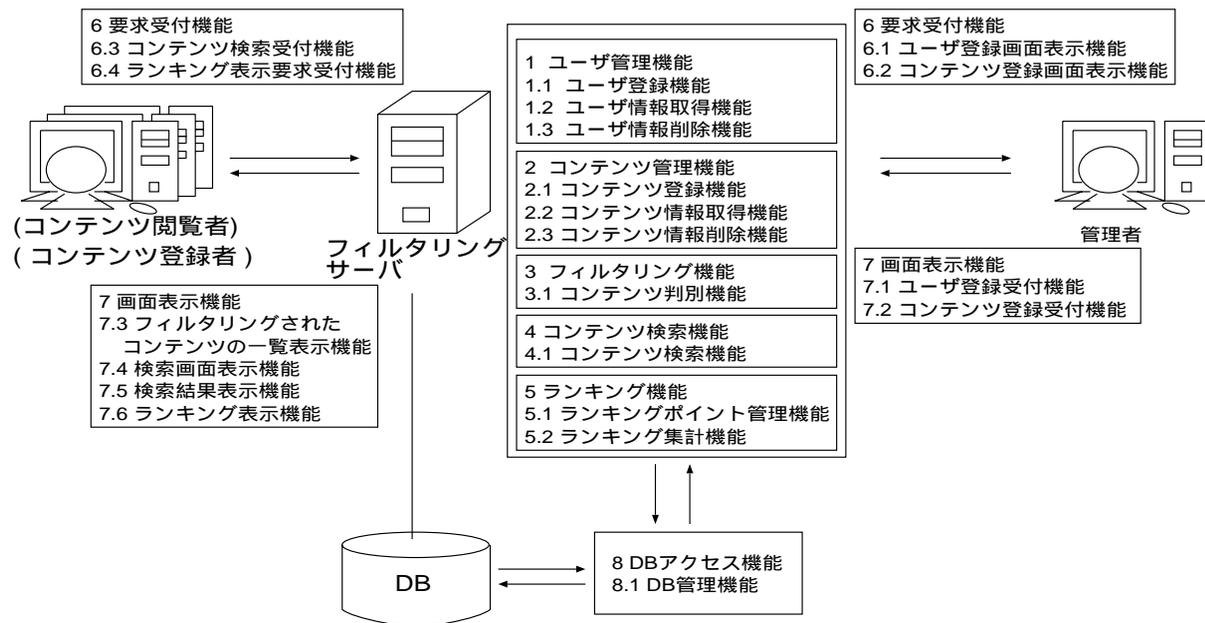


図 5.1 実験環境

## 5.2 実験環境の構築

実験環境を構築する上で、プロファイルをデータベースに蓄積するために、Postgresql7.0.2 を使用し、フィルタリングとユーザインタフェースの実装には、Java1.2.2 を採用した。また、DB と Java1.2.2 を結ぶインタフェースは、JDBC を使うことにした。この節では、Postgresql7.0.2、Java1.2.2、JDBC の概要とその有用性について述べる。

### 5.2.1 Postgresql

Postgresql は、カリフォルニア大学バークレー校 (UCB) で開発された研究用データベース管理システム「postgres」から発展した完全フリー、オープンソースのデータベース管理システムである。Postgresql7.0.2 は、移植性が高く、大部分の UNIX プラットフォームで動かすことのできるデータベースである。商用目的も含めて無償で利用でき、完全なソースコードが公開されている。データベース用の標準言語である SQL をサポートし、C 言語をはじめ C++,Java,Perl,Tcl/Tk,Python などの多彩なプログラミングインタフェースに利用できる。

Postgres を使用した理由は、フリーソフトウェアであり、ドキュメントが揃っているという点で採用した。

### 5.2.2 Java2

Java2 とは、Java Development Kit 1.2 (JDK1.2) のことである。

Java 言語の主な特徴は、以下の通りである。

- オブジェクト指向プログラミング言語
- 複数の仕事を同時に行うマルチスレッド処理が可能な言語

Java を使用した理由としては、開発するにあたり、クラスライブラリが豊富な点とドキュメントが揃っているので開発が用意であることから、利用した。

## 5.2 実験環境の構築

### 5.2.3 JDBC

JDBC は、Java で書かれたアプリケーションやアプレットがリレーショナルデータベースにアクセスできるように用意された Java クラスの集まりである。

JDBC には、2つのインタフェースが組み込まれている。

1. JDBC API(アプリケーションインタフェース) 2. JDBC ドライバ  
今回、作成したサーバのコンソール上から起動する Java アプリケーションは、JDBC API を使用している。もうひとつの、Java アプレットは、Postgres に用意されている JDBC ドライバを使用している。

Postgres に付属する JDBC ドライバは、「ダイレクトドライバ」と呼ばれるもので、JAVA だけで記述されており、しかも ODBC などに頼らず直接データベースに接続できるタイプである JDBC の構成図については、5.2 に示す。

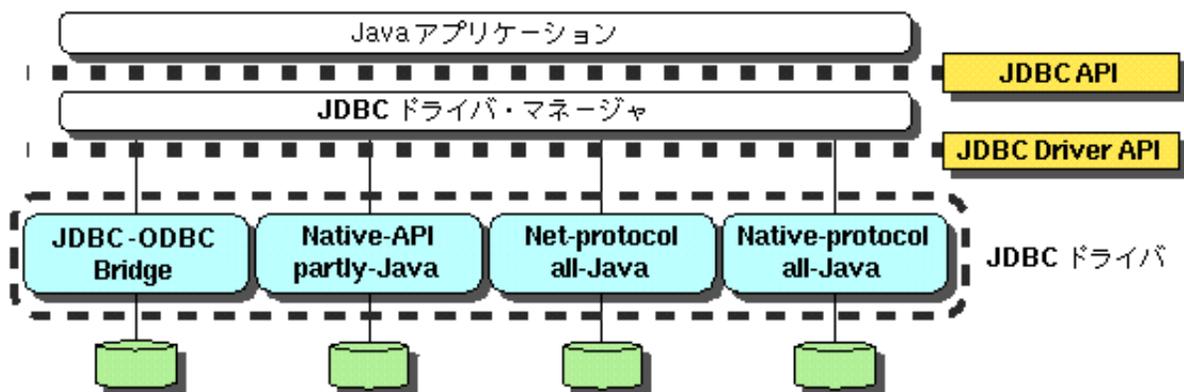


図 5.2 JDBC

今回使用したタイプの利点は、native メソッドを含まないこと。によって、アプレットから利用が可能になる。プラットフォームを選ばず可搬性が高いことがあげられる。欠点としては、クラスライブラリが大きくなりがちなことである。

アプレットからも、利用可能であるため、今後 WEB 上で実験を行うことを考慮している。

## 第 6 章

# 静的プロフィール項目への登録

### 6.1 静的プロフィール項目の登録

ここでは、静的ユーザプロフィールと静的コンテンツプロフィールへの登録機能について説明する。

登録機能は、フィルタリングを行う際の値になるものである。

## 6.1 静的プロフィール項目の登録

### 6.1.1 静的ユーザプロフィール

静的ユーザプロフィール登録処理の流れを 6.1 で示す。

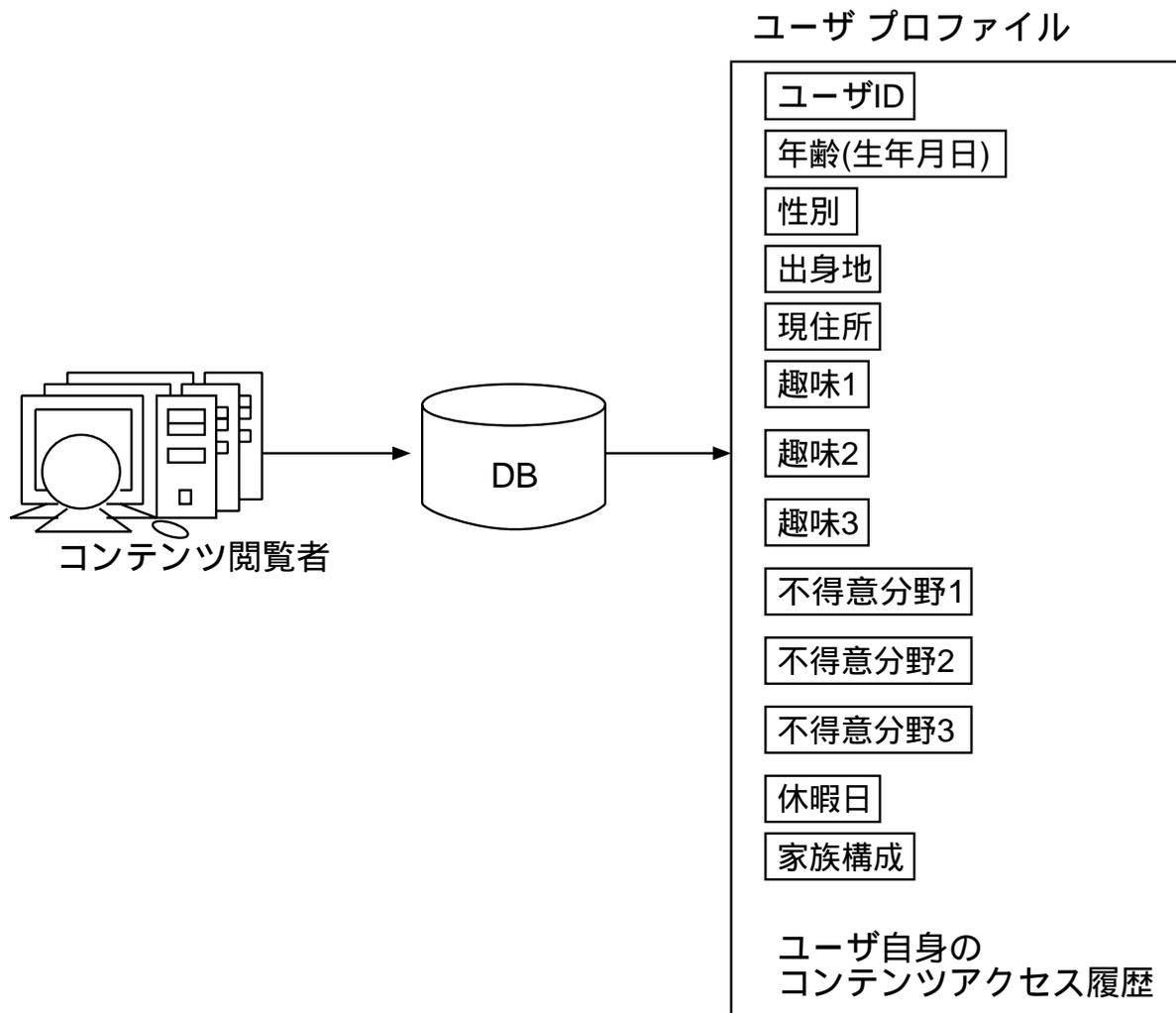


図 6.1 静的ユーザプロフィール登録処理の流れ

## 6.1 静的プロフィール項目の登録

### 6.1.2 静的コンテンツプロフィール

静的コンテンツプロフィール登録処理の流れを??で示す。

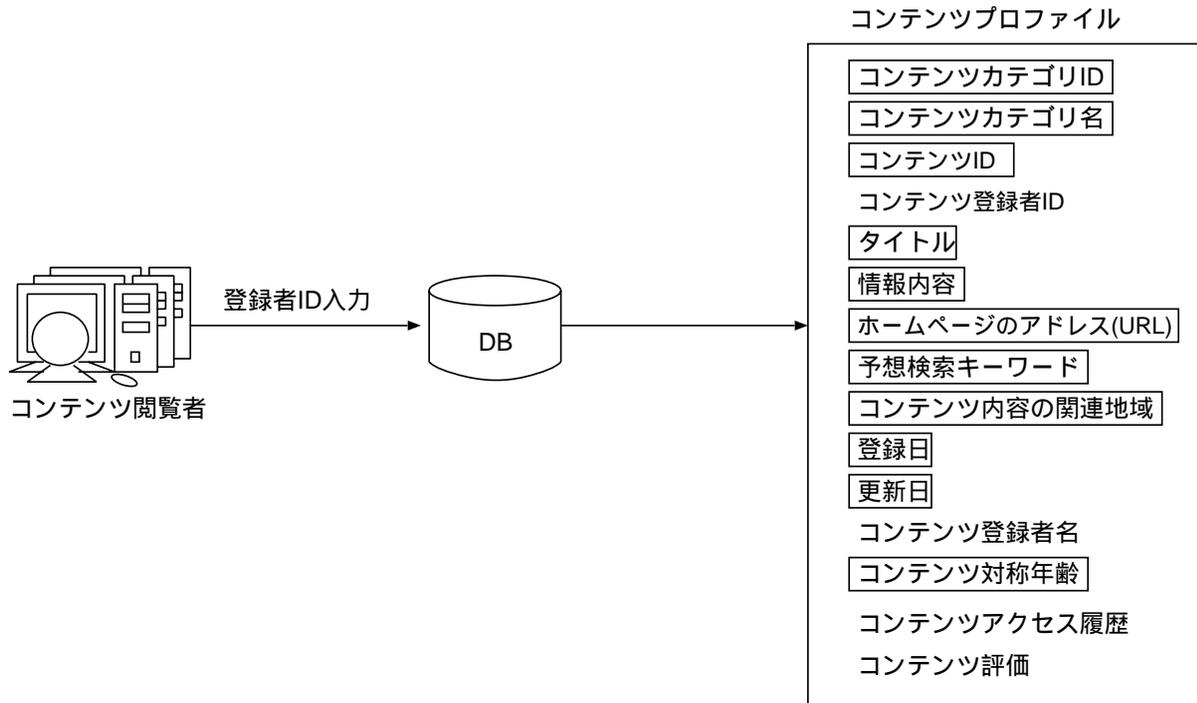


図 6.2 静的コンテンツプロフィール登録処理の流れ

## 第7章

# まとめと今後の展望

本研究では、様々な状況に適応できるフィルタリング技術を提案し、実験環境を構築を行った。今後、動的プロフィールの蓄積、更新方法を利用するフィルタリングアルゴリズムについて検討する必要がある。

さらに、各ユーザ、各コンテンツ項目のシステムにおける有効性を検証しなければならない。そのために、ユーザとコンテンツ間のフィルタリング実験を繰り返すことで各プロフィール項目の相関を調べ、ユーザに対して的確なコンテンツ選別のできる方式を考察しなければならない。また、その他の問題点として、コンテンツはこのシステムにコンテンツ作成者が登録する形態を取っていることから、コンテンツの自動認識化と自動プロフィール項目の登録などの検討も行わなければならない。

### 7.1 動的プロフィールの蓄積，更新方法

今後フィルタリングを行う際に、動的プロフィール無い。このシステムの威力を発揮することができない。

現在、考えられる動的プロフィール蓄積は、ユーザがコンテンツにアクセスするときに蓄積するようにする。一つのコンテンツに多数のユーザが、アクセスするとしよう。そのコンテンツにアクセスするユーザのプロフィール項目の一部をコンテンツプロフィール側で、蓄積する。このことにより、蓄積したプロフィール項目の分析が行える。例えば、コンテンツのユーザアクセス傾向を導き出し、プロフィール項目の似たユーザが、システムを利用する場合に対象するユーザに、そのコンテンツを配信をすることも可能である。

## 7.2 初期設定プロファイルの検討

動的プロファイル蓄積項目と蓄積方法の図は 7.1 で示す。

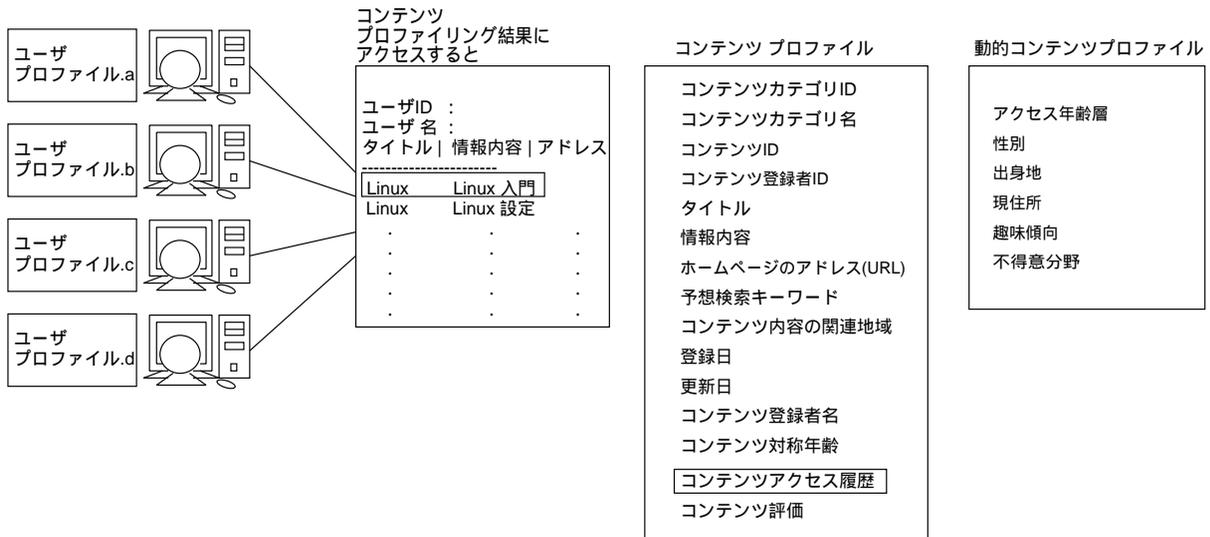


図 7.1 動的プロファイルの蓄積例

## 7.2 初期設定プロファイルの検討

絶対処理を行い関係がありそうな項目間のフィルタリング結果を基に、有用な項目と不要な項目について実証実験より明らかにしなければならない。また、その結果を利用したフィルタリングも開発しなければならない。

## 7.3 フィルタリングアルゴリズムの改良

今後、動的プロファイルを使ったフィルタリングを繰り返すことによって、プロフィール項目の変更なども予想される、このことによって必然的にフィルタリングアルゴリズムの改良も行わなければならない。

## 7.4 付加機能

本検討の実現にあたり，従来技術で述べた情報検索システムやブックマークエージェントも付加機能として搭載する必要性も考えられる．また，コンテンツのカテゴリごとによって，アクセスの頻度の高いコンテンツを紹介するランキング機能についても検討しなければならない．ランキング表示は，一週間ごともしくは，一月ごとの表示にする可能性もある．

また，ランキング上位のコンテンツを分析し，人気のあるコンテンツの型をコンテンツ登録者に配信し，よりよいコンテンツの作成を促せたい．

# 謝辞

高知工科大学情報システム工学科 清水 明宏 助教授には，本論文をまとめるにあたり，懇切丁寧なる御指導，御鞭撻を賜った．ここに謹んで深謝申し上げます．

また，本研究室 井上 富幸 氏 ，岡田 実氏 ，田鍋 潤一郎氏には，貴重な御教示を頂いた．特に岡田 実氏には，公私にわたり，御指導を賜り，本論文に対して御助言を頂いた．諸氏に心から御礼申し上げます．

本研究室学部生 伊藤 雄君 ，大石 恭裕君 ，岡崎 友輝年君 ，小橋 誠治君 ，篠原直之君 ，竹内 紀貴君 ，谷藤 喜彦君 ，林 竜也君 ，安岡 隆司君 ，山崎 愛さん に感謝します．さらに在学中，親切なる御助言を頂いた諸先生方に心より感謝します．

## 参考文献

- [1] 森田 昌宏, 速水 治夫, ”情報フィルタリングシステム, ”, 情報処理 ,vol.37,no.8,pp.751-757,1996.
- [2] 森, 他, ”ブックマークエージェント: ブックマークの共有による情報検索の支援”, 電子情報通信学会, Vol.J83-D-1 No.5 pp.487-494 ,2000 年.
- [3] 田中, 他, ”利用者に合わせた情報ナビゲーションに関する検討”, 電子情報通信学会, .
- [4] 森, 他, ”ブックマークエージェントによる WWW の URL 情報共有”, 情報処理学会, .
- [5] 渥美雅保/Web ページからのユーザの興味の遺伝的アルゴリズムに基づく抽出, 情報処理学会研究報告. ICS, [知能と複雑系], Vol. 97 Num. 51 pp.13-18 (1997.05) , , ”” , .
- [6] 沼尾, 横山/階層化された知識の継承による情報フィルタリング, 情報処理学会研究報告. ICS, 知能と複雑系, Vol. 99 Num. 47 pp.43-48 (1999.05)
- [7] 船越, 大黒/精度指向のハイブリッド情報フィルタリングの提案, 情報処理学会研究報告. ICS, 知能と複雑系, Vol. 99 Num. 63 pp.127-134 (1999.07)
- [8] 森田, 速水/情報フィルタリングシステム: 情報洪水への処方箋, 情報処理, Vol. 37 Num. 8 pp.751-758 (1996.08)
- [9] 溝口, 大和田他/帰納学習に基づく電子メールエージェント: 帰納学習によるアプローチ, 情報処理学会全国大会講演論文集, Vol. 第 52 回平成 8 年前期 Num. 1 pp.91-92 (1996.03)
- [10] 和田, 菅井/インターネット上の情報フィルタリング (1): 概要, 情報処理学会全国大会講演論文集, Vol. 第 51 回平成 7 年後期 Num. 4 pp.85-86 (1995.09)
- [11] 菅井, 和田/インターネット上の情報フィルタリング (2): 情報の整理方法, 情報処理学会全国大会講演論文集, Vol. 第 51 回平成 7 年後期 Num. 4 pp.87-88 (1995.09)
- [12] 森, 山田/ブックマークエージェントによる WWW の URL 情報の共有, 情報処理学会全国大会講演論文集, Vol. 第 54 回平成 9 年前期 Num. 3 pp.49-50 (1997.03)

## 参考文献

- [13] 宮原, 岡本/分散作業/学習環境を指向した協調フィルタリングシステムの開発, 情報処理学会全国大会講演論文集, Vol. 第 54 回平成 9 年前期 Num. 3 pp.297-298 (1997.03)
- [14] 菅井, 和田他/WWW 上の電子新聞に対する情報フィルタリング, 情報処理学会全国大会講演論文集, Vol. 第 53 回平成 8 年後期 Num. 3 pp.223-224 (1996.09)
- [15] 西, 中嶋他/ユーザの好みを考慮した新聞記事のランキング, 情報処理学会全国大会講演論文集, Vol. 第 55 回平成 9 年後期 Num. 3 pp.228-229 (1997.09)
- [16] 橘高, 鈴木他/個人適応型情報フィルタリングシステムの適用評価について, 情報処理学会全国大会講演論文集, Vol. 第 55 回平成 9 年後期 Num. 3 pp.234-235 (1997.09)
- [17] 北川, 中嶋他/個人属性を考慮した情報フィルタリング, 情報処理学会全国大会講演論文集, Vol. 第 56 回平成 10 年前期 Num. 3 pp.195-196 (1998.03)
- [18] 住田, 鈴岡他/WWW 情報フィルタリング・検索システム : FreshEye : サービス概要, 情報処理学会全国大会講演論文集, Vol. 第 57 回平成 10 年後期 Num. 3 pp.147-148 (1998.10)
- [19] 鈴岡, 澤島他/WWW 情報フィルタリング・検索システム : FreshEye : 全体システムの構成と動作, 情報処理学会全国大会講演論文集, Vol. 第 57 回平成 10 年後期 Num. 3 pp.149-150 (1998.10)
- [20] 加藤, 沼尾/2U-8 情報フィルタリングの手法を用いた転送先学習型メタ検索エンジン, 情報処理学会全国大会講演論文集, Vol. 第 58 回平成 11 年前期 Num. 3 pp.”3-157”-”3-158” (1999.03)
- [21] 宮原, 岡本/協調フィルタリングシステムにおけるエージェントの構成, 電子情報通信学会技術研究報告. AI, 人工知能と知識処理, Vol. 97 Num. 63 pp.15-22 (1997.05)
- [22] 宮原, 岡本/自律エージェント群による情報の協調フィルタリング : 協調フィルタリングのコンセプトとその実現法, 電子情報通信学会技術研究報告. AI, 人工知能と知識処理, Vol. 96 Num. 346 pp.17-24 (1996.11)
- [23] 宮原, 岡本/協調フィルタリングにおける個人の興味領域の表規とその獲得法, 電子情報通信学会技術研究報告. AI, 人工知能と知識処理, Vol. 96 Num. 453 pp.15-22 (1997.01)

# 付録 A

## A.1 静的フィルタリングプログラムソース

```
import java.io.*;
import java.sql.*;
public class filtering {
    public static void main (String [] arg) throws
ClassNotFoundException, SQLException, IOException
    {
Connection conn = null ;
Statement st = null ;
ResultSet rs1 = null ;
ResultSet rs2 = null ;
ResultSet rs3 = null ;
String url = arg[0];
String usr = arg[1];
String pwd = arg[2];
// データベースへの接続
try{
    Class.forName("org.postgresql.Driver");
} catch (ClassNotFoundException ex) {
    System.err.println("ERROR: Could not find the specified JDBC Dribver");
    ex.printStackTrace();
    System.exit(1);
}
try {
    conn = DriverManager.getConnection (url,usr,pwd);
}catch (SQLException ex) {
```

## A.1 静的フィルタリングプログラムソース

```
System.err.println("ERROR:Could not open DB connection.");
ex.printStackTrace();
System.exit(1);
}
/*****
// kojिन/usertable への検索, 実行と結果の受取り
*****/
try {
    st = conn.createStatement();
    // ユーザ ID の入力
    System.out.println(); // 改行
    System.out.println("学籍番号を入力してください。");
    System.out.println(); // 改行
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    String inputuserid = br.readLine(); //キーボードから入力した
                                                    文字列を
str1 に読み込み
    int num = Integer.parseInt(inputuserid); //文字列を変換して int 型の変数に読み込
    み
    rs1 = st.executeQuery("select userid,hobbyid,username from
                            usertable where userid = "+ inputuserid + " ");
}catch(SQLException ex) {
    System.err.println("ERROR: Could not excute the specified SQL statement.");
    ex.printStackTrace();
    System.exit(1);
}
// usertable 検索の受取りを表示
try {
    int userid,hobbyid;
    String username;
    // userid,hobbyid,username の表示
    System.out.println(); // 改行
```

## A.1 静的フィルタリングプログラムソース

```
        while(rs1.next()) {
userid = rs1.getInt("userid");
hobbyid = rs1.getInt("hobbyid");
username = rs1.getString("username");
System.out.println("ID number = "+ userid);
System.out.println("NAME = " +username);
System.out.println(); // 改行
        }
} catch (SQLException ex) {
    System.err.println("ERROR: Could not get result value.");
    ex.printStackTrace();
    System.exit(1);
}

/*****/
// kojinhobbyid への接続
/*****/
/*****/
// usertable の hobbyid を hobbytable に接続するために読み込む
/*****/
//kojin=# select * from hobbytable;
// hobbyid | hobbyname | update
//-----+-----+-----
//      1 | 音楽      | 2001-01-17
// kojinhobbytable の結果出力

String hid1 ;
hid1 = rs1.getString("hobbyid");
int num1 = Integer.parseInt(hid1); //文字列を変換して int 型の変数に読み込み
// hobbytable への検索, 実行と結果の受取り
try {
    st = conn.createStatement();
```

## A.1 静的フィルタリングプログラムソース

```
        rs2 = st.executeQuery("select * from hobbytable where hobbyid = "+ hid1 + " ");
    }catch(SQLException ex) {
        System.err.println("ERROR: Could not excute the specified SQL statement.");
        ex.printStackTrace();
        System.exit(1);
    }
    try {
        int hobbyid;
        String hobbyname;
        // hobbyid,hobbyname,update の表示
        System.out.println(); // 改行
        while(rs2.next()) {
hobbyid = rs2.getInt("hobbyid");
hobbyname = rs2.getString("hobbyname");
System.out.println( hobbyid + " : " +
        hobbyname);
        }
        System.out.println(); // 改行
    } catch (SQLException ex) {
        System.err.println("ERROR: Could not get result value.");
        ex.printStackTrace();
        System.exit(1);
    }
    /*****/
    // usertable の hobbyid を contentstable に接続するために読み込む
    /*****/
    String hid2 ;
    hid2 = rs1.getString("hobbyid");
    int num2 = Integer.parseInt(hid2); //文字列を変換して int 型の変数に読み込み
    // contentstable への検索, 実行と結果の受取り
    try {
        st = conn.createStatement();
```

## A.1 静的フィルタリングプログラムソース

```
        rs3 = st.executeQuery("select * from contentstable where hobbyid = "+ hid2 +" ");
    }catch(SQLException ex) {
        System.err.println("ERROR: Could not excute the specified SQL statement.");
        ex.printStackTrace();
        System.exit(1);
    }
// kojini/contentstable の結果出力
try {
    int hobbyid,contentid;
    String title,comment;
    String address;
    // contentid,hobbyid,title,comment,filename,regdate,update,reguser の表示
    System.out.println(); // 改行
    while(rs3.next()) {
contentid = rs3.getInt("contentid");
title = rs3.getString("title");
comment = rs3.getString("comment");
address = rs3.getString("URL");

System.out.println( contentid + " : " +
    title + " : " +
    comment + " : " +
    address);
System.out.println(); // 改行

    }
    System.out.println(); // 改行
} catch (SQLException ex) {
    System.err.println("ERROR: Could not get result value.");
    ex.printStackTrace();
    System.exit(1);
}
```

## A.1 静的フィルタリングプログラムソース

```
    }  
}
```