

## マルチエージェントに基づく情報検索システム

木原 英人<sup>†</sup>      岡田 ロベルト<sup>†</sup>      李 殷碩<sup>‡</sup>      白鳥 則郎<sup>†</sup>

<sup>†</sup>東北大学 電気通信研究所 / 情報科学研究科

<sup>‡</sup>韓国 成均館大学 工科大学 情報工学科

インターネット上には膨大な情報が存在し、その情報は増え続けている。この情報の海の中から求める情報を探すことは非常に手間と労力が必要である。たとえ Web ページを分野別に分類した索引サービスを使うにしても、検索エンジンを使ってキーワード検索をするにしても、ユーザが個々の操作をしていかなければならない。このような負荷を減らすために、ユーザのかわりに行動するエージェントを導入した。ここでは、エージェントが行う仕事を 2 種類に分けて考え、それぞれ別の種類のエージェントを作成した。これらのエージェントが複数存在して互いにやりとりしながら動作する、情報検索システムの構築を行った。本論文ではこのエージェント指向情報検索システムについて述べる。

## Information retrieval system based on multi-agents

Hideto KIHARA<sup>†</sup>      Roberto OKADA<sup>†</sup>      Eun-Seok LEE<sup>‡</sup>  
Norio SHIRATORI<sup>†</sup>

<sup>†</sup>Research Institute of Electrical Communication/Graduate School of Information Sciences, Tohoku University, Sendai, Japan

<sup>‡</sup>Department of Information Engineering, Faculty of Engineering, Sung-Kyun-Kwan University, Seoul, Korea

The amount of information existing on the Internet is huge and increasing very fast. In such world-wide network, finding the desired information requires that the user spends much time and effort. Even using existing facilities e.g. search engines based on index or catalogs like InfoSeek and Yahoo, the user has to perform the required operations. In this paper, we report on the multi-agent based information retrieval system we have implemented, in order to decrease the user load and increase quality of information.

## 1 はじめに

インターネット上には膨大な情報が存在する。その中から求める情報を探すには、かなりの手間と労力が必要である。たとえ、情報を分野別に整理した索引サービスを使うにしても、検索エンジンを使ってキーワード検索をするにしても、ユーザがいちいち操作していかなければならない。このようなユーザの負担を軽減することを目的として、ユーザのかわりに細かな操作を行うエージェントを使ったシステムの構築を行った。WWW(World Wide Web)上の情報検索を補佐するシステムである。

現在、WWW上で情報を検索する場合、BrowsingとSearchingという2つの方法がある。

Browsingの場合には、ユーザがリンク(Hyper Link)をたどって、行きつ戻りつしながらWWWページを調べていく。Searchingの場合は、ユーザはキーワードを入力して、そのキーワードが含まれるページのリストを検索エンジンから得て、そこからリンクをたどってページを調べていく。

いずれにしても、ユーザがいちいち操作しなければならず、目的の情報をみつけるまでに、退屈な操作を繰り返さなければならない、という問題がある。

## 2 マルチエージェントに基づく情報検索システム

### 2.1 概要

前述のようなユーザの負担を軽減するために、ユーザの代わりに細かな作業をしてくれるものが望まれる。この作業としては、たとえば、ある事柄に関する情報はどこにあるか？ということを実人に聞くことや、複数の情報源からこま切りの情報を得て、それをまとめて示すことなどである。

このような役割を果たしてくれるものとして、ユーザのかわりに行動するエージェントを導入する。

ここで考えるエージェントの性質は、以下の点である。エージェントどうして自分が知らない事をたずねあうことで、複数のエージェントの全体としての性能を上げる。たずねるエージェントを決めるときに、ほかのエージェントに関する信頼度を参考にする。この信頼度は過去の経験(ユーザからの

フィードバック)をもとに決める。信頼度を参照して、ある事柄についてはどのエージェントにたずねればよいかを決める。つまり、信頼度によってエージェント間の信頼関係を築き、これをもとにして、質問を送るのに適切なエージェントを選ぶ。これによって、よりよい検索結果を得ることを目的とする。

エージェントが行うことを次の2種類に分けて考え、それぞれ、ユーザエージェントとマネージャというエージェントが担当する。

- ① ユーザの代理人として、他人に質問を行う。
- ② 特定分野の専門家として、情報のとりまとめを行う。

システムの全体像は図1のようになっている。

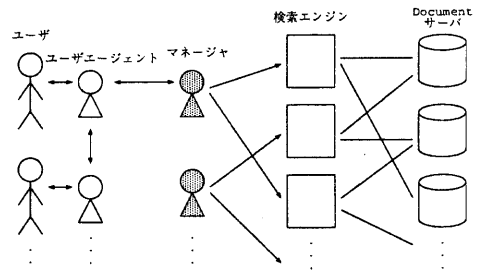


図1: エージェント指向システム

現段階の実装では、各エージェントは次のようなことを行う。① ユーザエージェントはユーザの代わりに他のエージェントに質問を行い、② マネージャは適切な検索エンジンを選び、検索を行う。

以下で各エージェントの役割について説明する。

- ① ユーザエージェントの役割: ユーザエージェントは各ユーザごとにひとつずつ存在し、ユーザの好みを学習して検索を補助する。ユーザの代わりに、他のエージェントに質問をすることによって、情報を探す。このとき、他のエージェントに対する信頼関係を築くことによって、ある場合にはどのエージェントが頼りになるかを把握する。それによって、より適切な情報へのポイントを定めることを目的とする。

ユーザに関するデータとして、ユーザがたどった場所を保存していく。ユーザが興味を持つ

た場所は、ユーザからのフィードバックによって把握する。また、ユーザからのフィードバックによって、他のユーザエージェントやマネージャに対する信頼度を更新する。

- ② マネージャの役割: マネージャは、いくつかの検索エンジンに対する信頼度を持っている。この情報によって、ある場合にはどの検索エンジンに聞けばいいかを決める。選択した検索エンジンに検索要求を送り、その結果を統合する。また、ユーザからのフィードバックをユーザエージェントから教えてもらうことにより、検索エンジンに対する信頼度を更新する。

## 2.2 エージェントの構造

ユーザエージェントとマネージャに共通する項目として、以下の点がある。

- 他のエージェントに関する信頼度のデータを保持し、ユーザからのフィードバックによって値を更新する。
- WWW サーバ側で動作する。
- 各プログラムは CGI(Common Gateway Interface) プログラムとして実装されている。

信頼度のデータは表1のようにになっている。各エージェントに関して、キーワードごとの信頼度と、そのエージェントそのものに対する信頼度(表1では“全体”という列)がある。

表 1: 他のエージェントに対する信頼度

他の agents	キーワード				
	全体	SF	soccer	fantasy	...
エージェント B	0.3	0.2	0.5	0.1	...
エージェント C	0.3	0.1	0.3	0.3	...
エージェント D	0.3	0.2	0.2	0.1	...
...	...	...	...	...	...

エージェント間のプロトコルには、WWW において用いられている HTTP(Hyper Text Transfer Protocol) を使う。

### 2.2.1 ユーザエージェントの構造

ユーザエージェントの構造は図2のようにになっている。ユーザエージェントが持つデータは以下の4つである。

**Topic:** ユーザが過去に入力したキーワード

**history log:** ユーザがたどった場所

**他の UA:** 他のユーザエージェントに対する信頼度

**Manager:** マネージャに対する信頼度

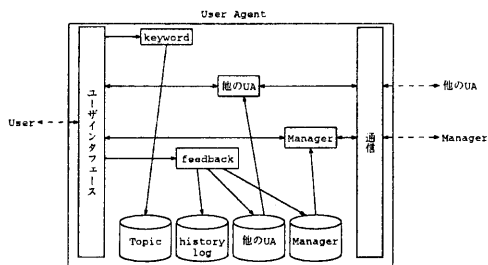


図 2: ユーザエージェントの構造

### 2.2.2 マネージャの構造

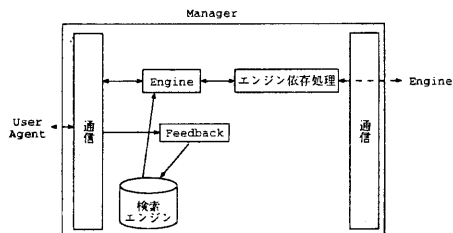


図 3: マネージャの構造

マネージャ(図3)は検索エンジンに対する信頼度のデータを持っている。これはユーザエージェントが持つ他のエージェントに関するデータと同じ形式である。

## 2.3 本システムの動作

### 2.3.1 エージェントの動作

実際に検索を行う場合の流れは次のようになる。

- (1) ユーザエージェントの起動
- (2) 他のエージェントへの問い合わせ
  - (a) 他のユーザエージェントへの問い合わせ
  - (b) マネージャへの問い合わせ

- (3) 結果の表示
- (4) ユーザからのフィードバックによる学習

以下で各段階についてひとつずつ説明する。

(1) ユーザエージェントの起動

ユーザは最初のページからユーザIDを入力することによって、自分のユーザエージェントを起動する。

ユーザエージェントは図4のように、WWWのサーバ側で立ち上がり、ユーザに検索キーワードの入力を促す。ユーザが過去に入力したキーワードを、関連するキーワード (Topic) として選択できるように表示する。

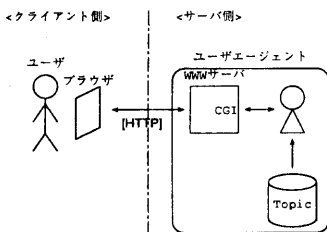


図4: ユーザエージェントの起動

(2) 他のエージェントへの問い合わせ

ユーザエージェントは、まず受けとったキーワードを、過去に入力されたキーワードのデータベースに追加する。次に、入力されたキーワードの検索を他のエージェントに依頼する。ここで、以下の2つの流れがある。ひとつは、(a) 他のユーザエージェントに聞くことで、もうひとつは、(b) マネージャに聞くことである。

(a) 他のユーザエージェントへの問い合わせ

他のユーザエージェントに聞く場合 (図5) は、入力されたキーワードに対して信頼度の高いユーザエージェントを選択し、選択したいいくつかのユーザエージェントに検索を依頼する。

図5の例では、ユーザは "soccer" というキーワードを入力している。信頼度の表の soccer の項目を見ると、ユーザエージェ

ント B(UA B) とユーザエージェント C(UA C) の信頼度が高いので、この2つのエージェントを選択する。

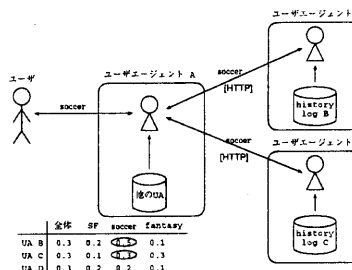


図5: 他のユーザエージェントへの問い合わせ

検索を依頼されたユーザエージェントは、自分のユーザがたどった場所を保存しているデータベース (history log) を参照して、キーワードのマッチングを取り、マッチしたページの場所 (URL: Uniform Resource Locator) を返す。このとき、そのページに対する重要度 (ユーザがそのページにアクセスした回数) も同時に返す。他のユーザエージェントから返ってきた URL を受けとったユーザエージェントは、それらの URL を見て、同じ場所が複数あればひとつにまとめてユーザに示す。

(b) マネージャへの問い合わせ

マネージャへ問い合わせを行う場合 (図6) は、ユーザから受けとったキーワードに対して信頼度の高いマネージャをひとつ選択し、そのマネージャに検索を依頼する。これは他のユーザエージェントを選ぶ場合と同じである。

検索を依頼されたマネージャは、検索エンジンに対する信頼度のデータを参照して、与えられたキーワードに対する信頼度の高い検索エンジンを選択する。選択した検索エンジンに検索を依頼して、返ってきた結果を統合して返す。

図では、"soccer" というキーワードに対して信頼度の高い検索エンジン2つ (Lycos と Alta Vista) を選択している。

(3) 結果の表示

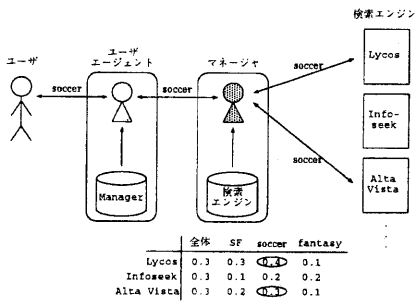


図 6: マネージャへの問い合わせ

他のユーザエージェントとマネージャからの検索結果が返ってきたら、ユーザエージェントは、それぞれの結果を表示する(図7)。この画面は、一番上の部分にユーザが入力したキーワードを示し、次の“from other User Agents”の部分が他のユーザエージェントから教えてもらったページへのリンクである。その下の“from Manager”の部分がマネージャからの結果である。

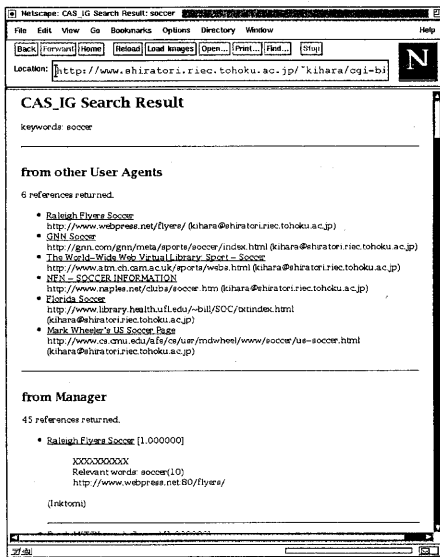


図 7: 結果表示画面

#### (4) ユーザからのフィードバックによる学習

結果のページからのリンクをユーザがたどると、ユーザエージェントは、指定された URL

の文書を取ってきてユーザに示すとともに、ユーザがたどった場所に関するデータベースにとってきた文書の URL などを追加する。

追加する事項は以下の6つである。

URL, タイトル, 日付, 重要度, キーワード, Topic

ここで、重要度は、基本的にそのページへの訪問回数で、ユーザからのフィードバックで更新する。また、キーワードと Topic はそれぞれ、ユーザが検索時に入力したキーワードであり、関連するキーワードである。

また、とってきたページをユーザに示すときに、ページの一番上にユーザからのフィードバックを受けとるコマンドを追加して表示する。このコマンドとしては、Positive Feedback と Negative Feedback と Exit Logging History の3つがある。

ユーザはいま示されているページが有用だと思えば、Positive Feedback を選択する。すると、ユーザエージェントは、history log の重要度の値を +n して重要度を上げ、次に、以下のように信頼度の表の数値を更新する。そのページが他のユーザエージェントから得られたページならば、そのユーザエージェントに対する信頼度を + $\alpha$  し、そのユーザエージェントの指定された Topic に関する信頼度を + $\beta$  する。また、そのページがマネージャから得られたページならば、マネージャに対する信頼度を更新するとともに、マネージャにフィードバックを伝える。

例えば、図5の例において、ユーザエージェント B から教えてもらったページで Positive Feedback を選択すると、信頼度は表2のようになる。つまり、ユーザエージェント B に対する信頼度を 0.3 から 0.4 に上げ、ユーザエージェント B の soccer に関する信頼度を 0.5 から 0.6 に上げる。

#### 2.3.2 新しいユーザエージェントの作成 — 掲示板エージェント

ユーザエージェントとしてのプログラムはひとつだけ存在し、各ユーザごとのデータを持つこと

表 2: 他のエージェントに対する信頼度

他の Agents \ キーワード	全体	SP	soccer	fantasy	...
ユーザエージェント B	0.4	0.2	0.6	0.1	...
ユーザエージェント C	0.3	0.1	0.3	0.3	...
ユーザエージェント D	0.3	0.2	0.2	0.1	...
...	...	...	...	...	...

によって、別々のユーザエージェントとしている。従ってユーザエージェントの数は増やすことができる。

ユーザエージェントが受けとったユーザ ID が全く新しいものであれば、新しいユーザエージェントをそのユーザに対して作成する。

このとき、どのようなユーザエージェントやマネージャがいるかを掲示板エージェントに教えてもらう。この掲示板エージェントは、どのユーザエージェントもその存在とアクセス方法を知っている、というようなエージェントである。この掲示板エージェントを介して、エージェントどうしが知り合うことになる。

### 3 問題点と課題

現在のシステムは、本当に基本的な機能しかない。もっとユーザの負荷を軽減する機能が必要である。

本システムで本当にユーザの負荷が軽減できるのか、という評価に関してはこれからの課題である。少なくとも、このシステムを使うことによって、ユーザがいちいち操作して求める情報を探すという負荷に関しては軽減できるだろう。

ユーザが過去に検索したキーワードをそのまま Topic として保存しているが、検索を繰り返すにつれて、非常に膨大なものになる。関連するキーワードをまとめて何らかのカテゴリを作成するなどの方法が必要である。

### 4 まとめ

WWW 上の情報検索において、ユーザの負荷を軽減することを目的として、エージェントを導入したシステムの実装を行った。

このシステムの目的は、以下の 2 つである。

1. ユーザがいちいち操作しなくても、求める情報を得ることができるようにする。
2. 得られる情報をより適切なものにする。これは、ある事柄についてはどのエージェントに聞けばいいかというエージェント間の信頼関係を築き、より適切なエージェントを選択することで行おうとしている。

将来的な目標としては、各エージェントをもっと賢くすることを考えている。マネージャは特定分野におけるエキスパートとして、複数のサーバから情報を寄せ集めて、ユーザの求める情報を構築する、というような機能を目指している。たとえば、旅行の予約において、飛行機のチケットの手配や、ホテルの予約を行い、旅先の地図などの情報をまとめてユーザに示してくれるような機能である。

また、サーバ側にもエージェント (マシンエージェント) を導入して、特定の人には見せるが、他の人には見せないなどの、情報のアクセス権限の制御を行ったり、サーバの文書が変更されたら、それをマネージャに伝えたりすることによって、情報の洗練をはかることも考えられる。

### 参考文献

- [1] R. Okada, E. Lee and N. Shiratori: "A Society of Cooperative Agents on the Information Network: Towards Intelligent Information Gathering" in Proc. 1995 Int'l Conference on Network Protocols
- [2] Yezdi Lashkari, Max Metral, and Pattie Maes: "Collaborative Interface Agents", Proceedings of AAAI '94 Conference, August 1994
- [3] Erik Selberg and Oren Etzioni: "Multi-Service Search and Comparison Using the MetaCrawler", Proc. International Conference WWW'95, Dec 95
- [4] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell: "WebWatcher: A Learning Apprentice for the World Wide Web", 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, March 95