

「地域福祉エージェントシステムの意義と諸機能に関する実証的研究」

松島桂樹⁽¹⁾
清水則茂⁽²⁾
新原健一⁽²⁾
杉塚弘泰⁽²⁾
大塚一
佐藤亘⁽³⁾

はじめに

平成10年10月に「大垣地域福祉情報ネットワーク研究会」(株式会社セイノー情報サービス、タック株式会社、河合石灰工業株式会社、大垣共立銀行株式会社、太平洋工業株式会社、岐阜経済大学による共同研究コンソーシアム)が設立され、通信放送機構からの委託研究、ソフトピアジャパンとの共同研究のための推進組織として共同研究を進めてきた。

本共同研究プロジェクトの意義は、地域活性化の重要な組織形態として提唱される産・官・学連携のコンソーシアムを、ソフトピアジャパンの所在地である大垣において、今後、もっとも地域社会にとって重要な課題の一つである地域福祉へのITの適用をテーマとして組織されたことにある。そして、大垣市の代表的な情報サービス企業の経営者層の強力なスポンサーシップと、岐阜経済大学の全面的な協力によって研究活動が進められ、産・官・学の人的、情報面での連携は格段に向上したと思われる。

本論文は、その成果の一つとして、福祉エージェントシステムの意義と諸機能に関する実証実験について報告したい。

1. 研究の目的

高齢化および介護保険の導入など、地域福祉サービスをめぐる大きな変化に対応するために、新しい情報ネットワークの活用が重要な成功要因となると考えられる。本共同研究では、地域福祉情報ネットワークの構成要素として、エンドユーザーによる利用者層、中間サーバーを基礎とする地域イントラネット層、多様な情報源からなる福祉関連分散データベース層を識別したが、なかでも、地域情報にとって重要な要素である中間サーバーの主要機能としてエージェント機能に着目した。

本研究の目的は、このエージェント機能についての実証実験によって、その価値と可能性について検証することにある。

2. 福祉エージェントシステムの理論的フレームワーク

(1) 地域福祉ネットワークにおけるエージェント機能の役割

地域福祉情報ネットワークでは、地域における利用者と情報源とを効果的に結び付ける役割が重要とされる。中間報告書^①では、機能レイヤーを図1のように定義した。とりわけ、ユー

(1) 日本アイ・ビー・エム株式会社

(2) ピーアイシステム株式会社

(3) 三重コンピュータシステム株式会社

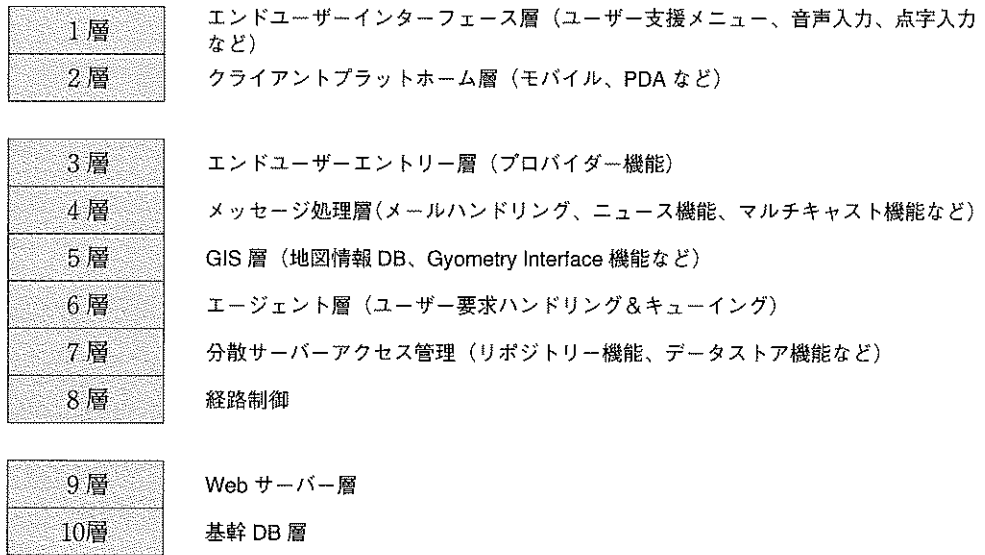


図1. 地域福祉ネットのレイヤー構造

ザーとサーバーとの間にある中間サーバー層を、地域イントラネットを介して、ユーザーの多様な情報要求に対応するために地域内および外部の情報源とを結び付けるレイヤーとして重要視した。

地域福祉情報システムのフレームワークでは、不特定多数を対象とするオープンな情報システムに加え、地域限定的な情報ネットワークが重要と考える。福祉サービスの受け手とサービスの提供者との間の高度な信頼関係を基に、ボランティアの意欲を高揚させるための情報ネットワーク構築が不可欠だからである。このような情報ネットワーク基盤は、まさしく地域に居住する個人と組織とを円滑に情報交換するためのクローズなネットワークと、世界の情報ネットワークとつなげるオープンエンドな性格とを併せ持っている。

インターネットの普及によって利用者の情報へのアクセスは飛躍的に高められたが、一般に情報源は分散的に管理されているため、利用者が膨大な情報の洪水のなかから必要な情報を得るために、各々のありか (URL) を探査しアクセスすることは実際には非常に時間と工数がかかる。そのため、利用者に代わって、利用者が求める情報源を探し自動的に情報収集するエー

ジェント機能が注目されている。

たとえばキーワードとして“介護”を登録しておけば日本中の関連情報を探してくれる。また、前もって利用したいサイトを登録しておけば、そこを優先的にアクセスする。もちろん、以前に収集した情報と変更なければ重複した情報収集を行なうことはない。収集済みの情報を中間サーバーに保存しておけば地域イントラネット内部の利用者はあらためてインターネットにアクセスする必要なく短時間に情報を入手できる。

とりわけ福祉ニーズは多様であり、さらにサービスの受け手も障害者や高齢者であったりする。それを標準的なユーザーインターフェースで取り扱うとするときわめて使いにくいシステムになってしまう。エージェント機能はまさしく福祉サービスの受け手と提供者との間に入って、提供者の側ではなく受け手の立場からサービスを創造する情報の窓口として機能する。まさしく、利用者の目的に沿ったメニューを用意することが多くの問題の解決を前進させる。たとえば福祉サービス提供者が事務所を出るときに、最新の関連情報をモバイル機器にダウンロードしておけば、出先から福祉施設の空き情報などをインターネットにアクセスするこ

となく現場で迅速に活用でき非常に効果的である。

さらに利用者の一元的な検索が容易になる。通常、市や県の福祉情報は市や県の情報の一部として提供されているため、福祉の情報集目的のために横断的にアクセスするのは必ずしも好都合ではない。すなわち、新しい情報システムには、地域割りや縦割り、サービスタイプごとの仕切りは不要なのであって、多様な福祉ニーズに対して必要な情報をサーバー横断的に組み合わせることで利用者に提供することが重要な使命となる。

また、地域イントラネット内部のサーバー、たとえば市や県の福祉情報などは、その複製情報(レプリケーション)を中間サーバーに蓄積しておくことによって自動的なバックアップとなる。

(2) 福祉エージェント機能の概要

地域福祉情報提供者とエンドユーザーの間におかれた中間サーバーの重要な機能の一つであるエージェント機能の地域福祉への適用可能性を、プロトタイプシステムをもとに実験し検証することが本研究の主要目的である。この実験のために、モバイルエージェントを使用した情報検索システムのプロトタイプを構築し、インターネットにおける情報検索システムの課題の解決を試みる。

今日、インターネット上に情報提供サイトは数多く存在し、独自の検索システムを提供している。ユーザーはそれらを利用して、様々な情報を入手することが可能であるが、それらの情報提供のメカニズムにはいくつかの課題を見つけることができる。

たとえば、

- ①情報提供サイトが多く、しかもそれぞれが独立して運営されている。その結果としてエンドユーザーは複数サイトの検索を行う際に、各サイトへそれぞれアクセスして検索を行う必要がある。
- ②既存の検索システムは基本的に同期的な処理を基に実現されている。すなわちエンドユー

ザーは検索リクエストを発行した後、結果が得られるまで待機していなければならない。また複数サイトを検索する際には、それぞれのサイトを順に同期的にアクセスすることになり、結果として時間的なコストは大きくなってしまふ。

- ③情報の洪水に拍車をかけ、必要な情報がどうでもよい情報に埋没されかねない状況を作り出し、逆に、検索の負担を大きくしている。

これらのため、いくつかのWebサイトでは、検索エンジンによる情報探索サービスを提供し、ある程度、この状況を改善しているが、時間、コスト等、まだまだ問題山積といった状態である。

一方、地域イントラネットにおいても、各情報提供サイトは独立して運営されているが、インターネットの場合と比較すると各サイト間の結びつきは大きい。したがって、利用者に対してポータルサイトを用意し、各情報提供サイトが協力して共通の情報提供システムを構築することによって、利用上の利便性を飛躍的に高めることができるであろう。

このような状況下で、最近、ソフトウェア技術の一つとして注目されつつあるエージェント、特に、検索作業に適したモバイルエージェントは上記の問題に対する解決策になりうるものとして期待されている。

今回の実証実験では、上記の課題点を解決するために、地域イントラネット内の複数の情報提供サイトを含んだ環境を想定して、モバイルエージェントを使用した情報検索システムの設

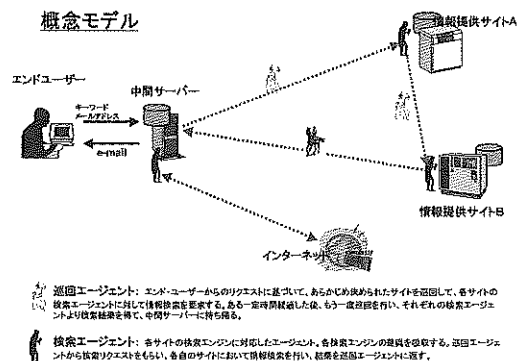


図2. 福祉エージェントの概念モデル

計を行い、そのプロトタイプを構築した。

(3) 福祉エージェント機能の概念設計

本研究における福祉エージェントは、モバイルエージェント、検索エージェント、巡回エージェントから構成される。

①モバイルエージェント

モバイルエージェントは、ネットワークに接続されたコンピュータ間を移動して、リモート・ホスト上で作業を行うプログラムである。モバイルエージェントの特長はその非同期性にある。あるプログラムが移動して、他のコンピュータ上で処理を行っている間、リクエスト側のコンピュータ上では他の処理を行うことができる。したがって各サイト間にまたがって非同期処理を行いたい場合に有効な手段である。特に情報検索システムの場合、検索を行っている間、回線を接続している必要がないため、ダイヤルアップ接続のように、ネットワーク接続に関する経済的コストが大きい環境では非常に魅力的である。エンドユーザーは情報検索のリクエストをサーバーに伝えた後、一度回線を切断し、必要な時にあらためて回線を接続して結果を得ることが可能となる。

モバイルエージェントの実現方法として、今回の実験では IBM 社のモバイルエージェント技術である Aglets を使用した。Aglets は、その実行環境やモバイルエージェントのプログラム自身が Java で記述されており、その可搬性の高さを考えて採用した。さまざまな検索サイトで実装ならびに実行されることを考慮すると可搬性を実現することは重要なことである。

②巡回エージェント

あらかじめ設定されている巡回リストに基づいて、エンドユーザーより入力されたキーワードを持ってネットワーク上の複数の情報提供サイトを巡回し、各サイトに常駐している検索エージェントと対話を行いそのサイトの検索結果を得る。また、集めた結果をエンドユーザーに返す。

③検索エージェント

検索エージェントは、情報を提供する各サイトごとに存在する。そして自分が存在するサイトに関する情報検索および情報提供についての処理を行う。自分の存在するサイトに訪れた巡回エージェントより情報検索のためのキーワードを受け取り、そのキーワードを基に検索を行い、巡回エージェントに対してその結果を渡す。検索エージェントが存在しないインターネット上のサイトを検索する場合の検索エージェントは中間サーバー上に導入する。

実証実験用の検索システムは、現実には即して各サイトごとに異なるものが実装されていることを想定し、検索エージェントは、巡回エージェントに対して、各サイトに実装されている検索システムの差異を吸収する役割を果たしている。したがって巡回エージェント自身は、自分が訪れた先の検索システムがどのようなものかに関する知識は不要である。巡回エージェントと検索エージェントの対話は、どのサイトを訪れているかにかかわらず、検索のためのキーワードと最大検索数をリクエストして、その結果を得るという非常に簡単なものである。

これらのエージェントは以下のような振る舞いを行う。

- ①エンドユーザーが、中間サーバーに用意された、リクエストのための HTML ページを参照する。このページ上で検索したいキーワードならびに結果を受取りたいメールアドレスを指定する。
- ②中間サーバー上では巡回エージェントが生成されて、エンドユーザーから入力されたキーワードをその属性として持ち、中間サーバー上に設定された巡回リスト（情報提供サイト A および B がリスト上に設定されていると仮定）にしたがって、巡回を始める。
- ③巡回エージェントは情報提供サイト A に到着して、A 上の検索エージェントを生成する。さらに生成された検索エージェントに対して、属性として持ってきたキーワードを基に検索を行うよう依頼する。巡回エージェント

は検索エージェントからの結果を待たずに次の情報提供サイト B へ移動を始める。

- ④情報提供サイト B に到着した巡回エージェントはサイト A と同様に、B の検索エージェントを生成して、検索を依頼する。その後巡回エージェントは一旦中間サーバーに戻る。
- ⑤一定の時間が経過した後、巡回エージェントは再び巡回リストに基づいて巡回を始める。サイト A およびサイト B を巡回して、それぞれの検索エージェントより結果を受け取り、中間サーバーへ戻る。
- ⑥巡回エージェントは受取った結果をまとめて、エンドユーザーから指定されたメールアドレスに結果を送信する。

このモデルの特長は情報提供サイト A ならびに B における検索処理が、並行して行われている点である。それぞれのサイトで検索に要する時間を T (サイト名) であらわすと、これらの処理を順次的かつ同期的に行えば、最終結果を得られるまでに要する時間は $T(A) + T(B)$ となる。それに対して、このモデルでは、サイト A での処理とサイト B での処理が並行処理されるために、最終結果を得られるまでの時間は $\text{Max}(T(A), T(B))$ となり、順次処理と比較して短い時間で結果を得られることがで

きる。

またエンドユーザーへの結果の提供も非同期で行われるため、エンドユーザーは検索リクエストを発行した後は、回線を一旦切断して他の業務処理を行うこともできる。結果を参照する場合には、自分のメールボックスを確認し結果を入手することができる。通常の検索システムのように検索結果を得られるまで、ブラウザーの前で待機している必要はない。このようにユーザーの業務効率を向上させることもこのシステムの特長の一つである。

(4) エージェントシステムと GIS (地理情報システム) との連携

エージェントは自動的に多くの情報を収集するが、それらが地図上のどこに存在するものなのかを示すものは、テキストで表現された住所情報のみである。しかし、テキストで示された位置情報だけでは、地理を詳しく知っている利用者以外にとっては不十分であるといえる。本研究では、GIS サーバーを用いて、エージェントが収集した情報と、その位置情報とを自動的に関連付けるためのシステムの構築を目指す。

1) アドレスマッチング

テキストで示された位置情報を GIS で扱う

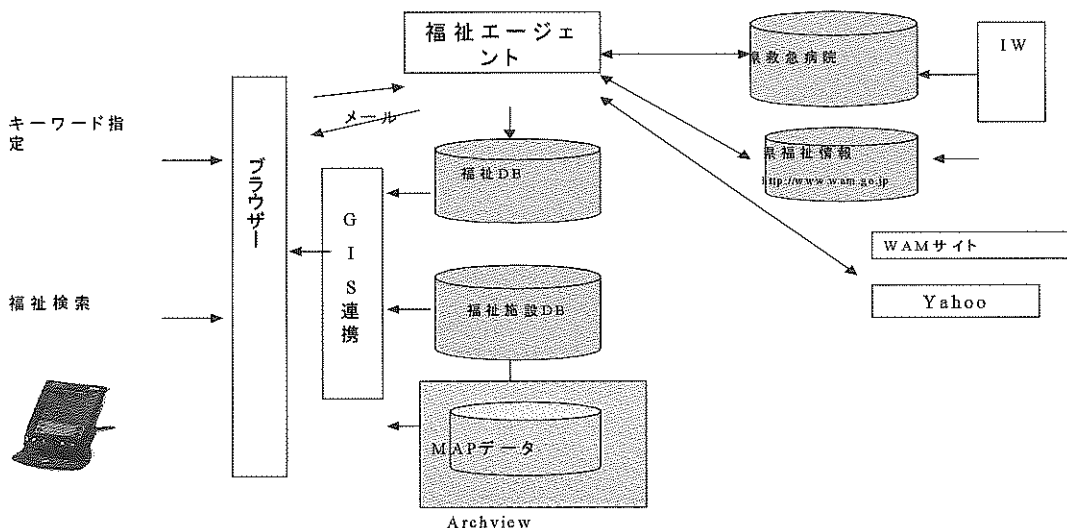


図 3. 福祉エージェントと GIS 関連図

ためにはアドレスマッチングが必要である。アドレスマッチングには、作成した地名辞書を参照し該当レコードの空間的位置情報（緯度経緯度座標）を返す方法と、既存の住所情報を持った地図データを利用する方法が考えられる。前者は手間がかかるが正確に登録することができる。また後者は、地図データに住所番地情報があれば簡易的に構築することが可能であるが、この登録内容が必ずしも正確とはかぎらないため、確認作業や手作業による補正が必要となる場合がある²⁾。両者を比較検討すると、本研究においては空間解析等の処理を必要としないため、位置情報の厳密さよりも地図データのセットアップおよび維持管理を安価かつ容易に行なえる後者の方法が妥当だと言える。

2) システム環境

システム環境は以下の表の通りである。

OS	Windows2000Server
WebServer	IIS5.0
GISアプリケーション	Arcview3.2
MapServer	Internet Map Server(Web上でのマップサービスに必要)
電子地図	ダイケイ 大垣市電子住宅地図

3) システム設計

システム設計として二案を検討した。第一案として、エージェントが巡回、収集した情報の中から位置をあらわす文を抽出し、既存の住宅地図の建物フィールドとアドレスマッチングを行なう。次にヒットした建物フィールドのレコードを地図表示するパラメータを取得し、エージェント側のDBにそのパラメータを格納する方法が考えられる。(図4参照) この方法では、地図表示の際に表示するためのパラメータが送られるだけであるため処理速度は速い。しかし、住所情報と建物フィールドのレコードの一致が複数あった場合の処理について設計上の考慮が必要である。

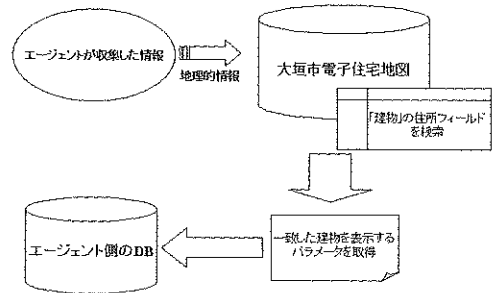


図4. システム設計案1

第二案として、位置をあらわす文を抽出し、キーとしてその住所情報をメールのリンクに付加(例: <http://210.248.52.100/.....key=大垣市北方町5番地5号/>)、地図表示要求があったときにGIS側でキーをもとに建物フィールド内の検索を行ない地図表示を行なう。この方法を採用する場合、住所情報と建物情報とを関連づけたリンクテーブルを作成していくことで処理の負担を軽減できると考えられる。また、建物と複数件がヒットした場合は、選択画面を出力しユーザーによる判断に頼るか、号レベルから番地レベルまで範囲を広げて位置を示すなどの対処をとることにする。

二つのシステム設計案を考慮した結果、前者はエージェント側で住所文字列の抽出、パラメータの格納などを行なうため作業が煩雑になることに加え、住所情報と建物情報との一致が複数あった場合に自動的に処理するのは困難であると考えられる。したがって後者が適していると考え、これを採用することにした。

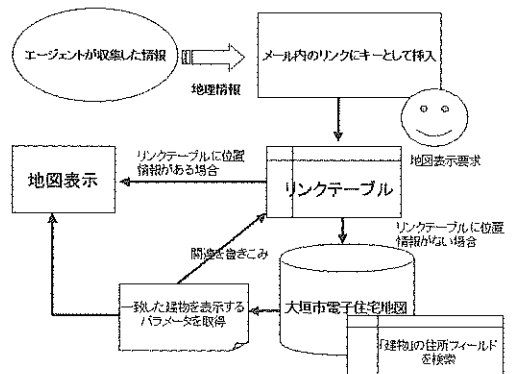


図5. システム設計案2

4) 課題

現段階では、エージェントが収集した情報の中から自動的に住所情報を抽出することができないため、手作業であらかじめ住所の文字列を抜き出し、実験検証を行なった。しかし、「茶筌」などの前文検索ソフトと地名辞書を組み合わせたり、「芭蕉」⁹⁾をツールとして用いて住所らしき文字列を抽出することも、先行研究にて試みられている。今後の研究課題としたい。

3. 実証実験概要

(1) エージェントシステムの開発環境

今回の実験を行ったシステムの構成図を示す。当システムの開発にあたって、次のような環境を整備した。

- サーバー 1：岐阜経済大学 Linuxサーバー 1、OS は Turbo Linux 6.1、Web システムは Apache1.3.14 を使用した。
- サーバー 2：岐阜経済大学 Linuxサーバー 2、OS は Turbo Linux 6.0、Web システムは Apache1.3.12 を使用した。

サーバー 3：岐阜経済大学 GISサーバー、OS は Windows2000Sever、Web システムは IIS5.0 を使用した。

また Linux サーバー 1 についてはソースからインストールしたため最新のバージョンの Apache1.3.14 を導入した。

システムの設計方針として、特定のハードウェアやオペレーティング・システムに依存しないことを考慮した。中間サーバーおよび情報検索サイトのオペレーティング・システムとしては、現在注目を集めている Linux を使用した。Linux は Web サーバーのオペレーティング・システムとしては、世界中でもっとも大きなシェアを持っており、今後もその使用実績は増加すると見られている。エンドユーザーへのインターフェースとなる Web サーバーも、使用実績のある Apache サーバーを使用した。

また前述したようにモバイルエージェント技術として、Aglets を用いた。Aglets は基本的に Java が稼動する環境で実行可能であり、Linux の他に Windows NT や AIX 等で使用することができる。各情報提供サイトの検索システムとしては、Linux 上で標準的に使われている

実験システムの構成図

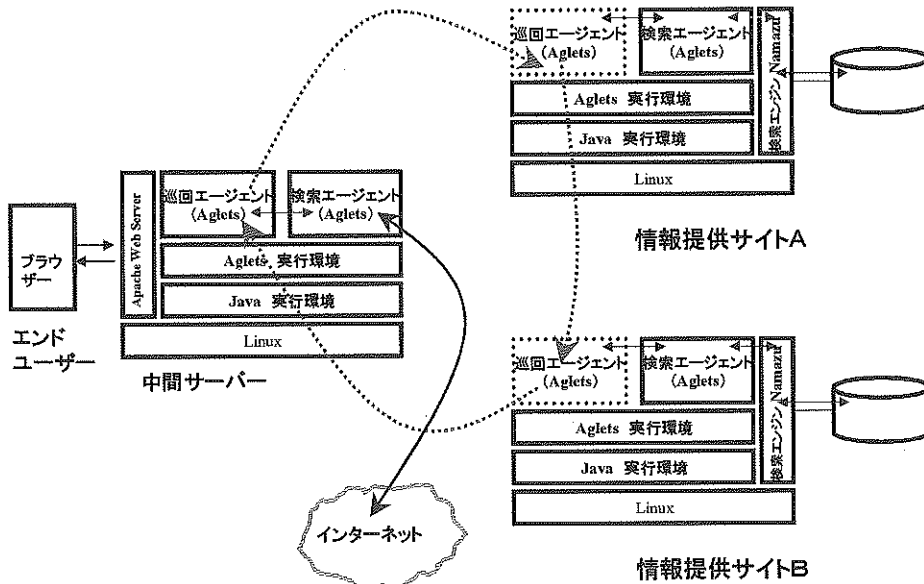


図 6. 実験システム構成図

Namazu というシステムを導入した。Namazu はあらかじめインデックスを作成しておき、そのインデックスに基づいて検索結果を返すタイプのシステムである。

このように特定の製品、ベンダーに依存しないシステムを提供することは、地域イントラネットのように、それぞれのサイトが独自のシステム構築を行っているような場合には特に重要なことである。また今回は、プロトタイプシステムには実装しなかったが、Yahoo や Infoseek のようなインターネットの検索システムにも対応することができる。このようなサイトには検索エージェントを導入することができないが、それぞれのサイトに対応した検索エージェントを中間サーバー上に導入して、それらの検索エージェントが Yahoo や Infoseek の検索インターフェイス、多くの場合は CGI インターフェイスをエミュレートすることにより、検索結果を得ることができる。巡回エージェントは、対話を行う検索エージェントが地域イントラネット内のサイトに対応した検索エージェントなのか、インターネット検索システムに対応した検索エージェントなのかを意識する必要はなく、透過的に処理を行うことができる。

(2) エージェントサーバの基本構成

開発運営にあたって、下図のようなシステム構成とした。

当システムは本学と IBM SE 研究所との共同研究により作成された。エージェントシステムは本体である `aglets.class` と周辺プログラムのまとまった `Norry.tar` から構成される。

1) `aglets1.0.3.class`

エージェントプログラム本体であり、インストール時点で使用する。Xwindow 上で、起動する必要があるため、X が動く環境が必要である。また、JDK のバージョンは 1.1.7 を使用した。

2) `Norry.tar`

① `KickAglet.class` : `QueryAglet` と `TravlerAglet` を起動する。デフォルトでは `/home/httpd/aglets/` の中にコピーする。

② `QueryAglet.class` : 全文検索システム (NAMAZU) に、検索するキーワード等を渡す。`Aglets` をインストールした後にできるフォルダ、`Aglets1.0.3/public/` の中にコピーする。

③ `TravlerAglet.class` : 指定されたサーバー間の橋渡しをする。`QueryAglet` 同様にこちらもイ

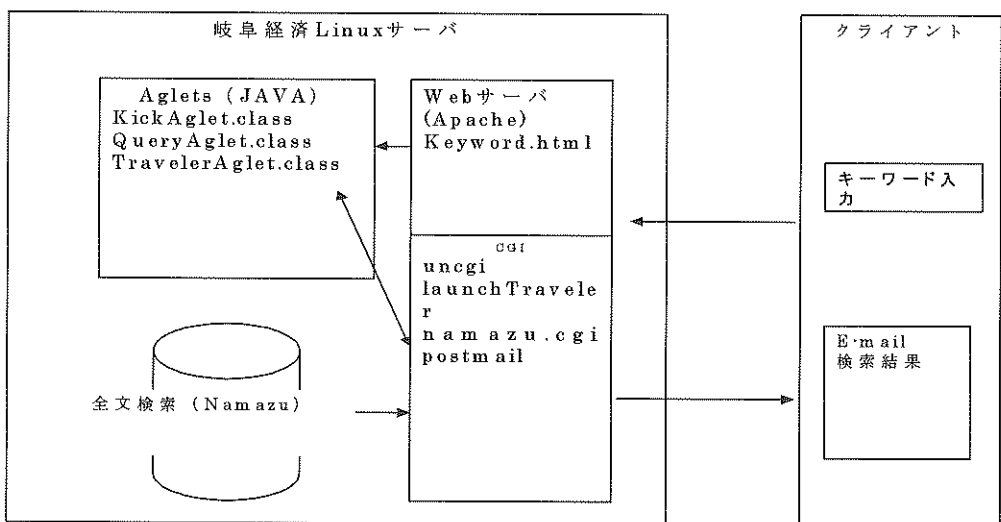


図7. サーバーの基本構成

インストール後にできる、Aglets1.0.3/public/の中にコピーする。

- ④Uncgi: ホームページで入力されたキーワードの処理をして、launchTravelerに渡す。こちらもCGIディレクトリにコピーする。
- ⑤LaunchTraveler:uncgi から渡されたキーワード等 KickAglet に渡す。また訪問するサーバー指定や、ファイルのPATH等環境設定をする。これもCGIディレクトリにコピーする。
- ⑥namazu.cgi: 全文検索を Web 上で行う CGI
- ⑦Postmail: 返信先アドレスにメールを送るスクリプト。これも、CGIディレクトリにコピーする。
- ⑧Keyword.html: クライアントがキーワードを入力するための、フォームの雛型。Webに公開できる場所に保存する。

(3) エージェントシステム構築の手順

1) Aglets 動作環境の構築

(ア) JAVA のインストール

#tar xvf norry.tar
と入力して圧縮されているファイルを解凍する。norry.tarを展開してできたフォルダの中に入っているjdk 1.1.7bを展開して、適当なフォルダにコピーする。一般的な環境では/usr/local以下となるので、このフォルダに解凍した。この後PATHを通して終了した。

(イ) Web サーバーと CGI の設定。

使っているデストリビューションにもよるが、Linuxには標準で入っているの、それを使用した。あわせてcgiも使えるよう設定した。

2) Aglets のインストールと設定

(ア) Aglets のインストール

aglets1_0_3.class を起動させる。インストールは Xwindow 上で行う。

```
#java Aglets1_0_3
```

(イ) CLASS のコピー

norry.tar の中にある3つのCLASS

をフォルダにコピーする。

```
#cp KickAglet.class /Aglets1.0.3/public
```

```
#cp TraverAglet.class /Aglets1.0.3/public
```

と入力して、Aglets をインストールしたディレクトリ以下の public のなかに2つのファイルをコピーする。

```
#mkdir /home/httpd/aglets
```

としてフォルダを作り、

```
#cp KickAglets.class/home/httpd/aglets
```

と入力して KickAglet.class をその中にコピーする。

(ウ) 初期設定

インストールしたディレクトリに Agletsv という起動スクリプトがあるのでこれを使って起動させる。

```
#/Aglets1.0.3/agletsv
```

起動したら初期設定をする。Option の Security Preferences アクセス権の設定などを行う。利用者の IP の許可とフォルダの書き込み権限の許可の設定する。trusted untrusted とともに File System の Read と Write の両方に/tmp以下を許可する設定をする。また、『networkにConnect』にサーバーのIPとポート番号80を加える。

(4) エージェントの基本動作

- ① keyword.html に入力されたキーワード「検索キーワード・ホスト名・電子メールアドレス・要求検索結果数」を UNCGI が抜き取って KickAglet を起動させる。
- ② KickAglet は TravelerAglet に情報を処理して渡す。
- ③ TraverAglet は指定されたサーバーに情報を送る。
- ④ 情報もらったサーバー内の QueryAglet が namazu.cgi を起動する。
- ⑤ namazu.cgi がデータベース検索をして結果を HTML 形式で返す。

(5) エージェントシステムの稼働実験

1) 作業環境概要

OS・・・WindowsNT Server + SP6 a /
Windows2000Server

WebServer・・・IIS4.0 /IIS5.0

エージェントシステム構成

Aglets

ActivePerl - 5.6.0.6

Namazu for Win32 - 2.05

Kakasi - 2.3.2

2) 環境構築手順

- ① WindowsNT Server/Windows2000 Serverをインストール。
- ② JDK1.1.7bのインストール。コマンドプロンプトから「java Aglets 1_0_3」と入力しagletsをインストール。あとは指示にしたがって作業を進める。
- ③ ActivePerlのインストール、<http://www.activestate.com/>にあるActivePerlのインストーラが自動的に.pl, .plxの関連づけ(マッピング)をして、システム環境変数PATHの登録も行う。
- ④ kakashi for Win32のインストール、インストール先はC:\kakashiとする。Kakashiを単独で使用する場合でなければ、この場所に展開するだけでよい。

- ⑤ Namazu for Win32ver2.05をインストール、インストーラがあるので実行する。この際インストール先はC:\namazuとする。インストールするとシステム環境変数のPATHも自動的に設定される。
- ⑥ IISの設定、「インターネットサービスマネージャ」を開き、「規定のWebサイト」を右クリックしてプロパティを表示する。ActivePerlがインストール時に「.pl」と「.plx」のマッピングを登録してくれるが、「.cgi」は登録されない。「.cgi」のマッピングを追加する。実行ファイルをPerl\bin\perlis.dll、拡張子を.cgi、スクリプトエンジン、ファイルの存在を確認のチェックボックスにチェックを入れる。
- ⑦ Namazu.cgiの設定、C:\namazu\libexec\namazu.cgi.exeこのファイルをC:\inetpub\scriptsにコピーする。

3) 確認シナリオ

- ① Agletsの起動確認。
「C:\Aglets1.0.3\agletsvr.bat」をダブルクリックして起動。初めての起動の際には図8のような画面で、登録を行う。2回目以降は図9のようなになる。

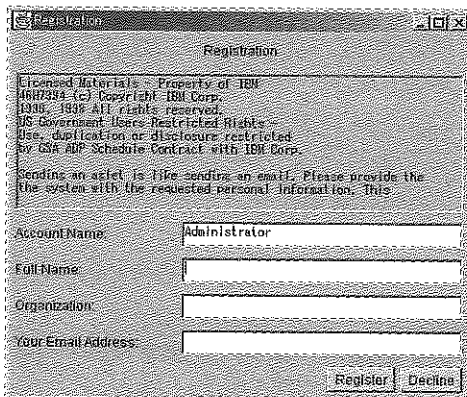


図 8

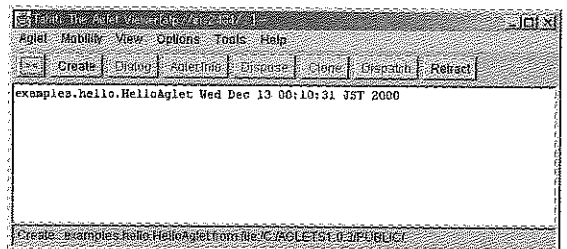


図 9

② Namazu の起動確認。

コマンドプロンプトで「C:\>namazu」と入力すると以下の図3のような画面になり起動が確認できる。

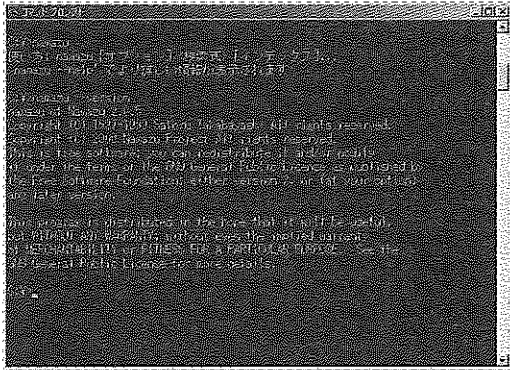


図10

③ Namazu.cgi の確認

ブラウザにて <http://210.248.52.102/scripts/namazu.cgi.exe> を開くと以下の図11のような画面になる。

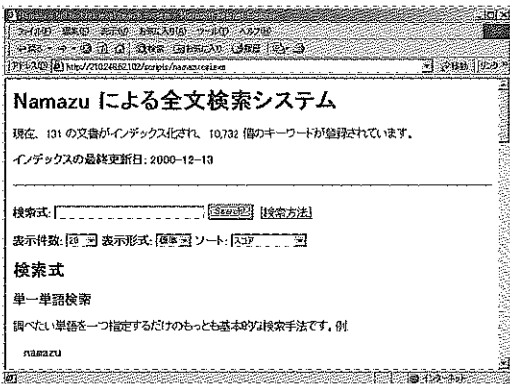


図11

4) 実験結果

エージェント機能のプロトタイプは、所期の環境にてほぼ成功裏に稼動した。とりわけ、Linux、WindowsNT などの複数の分散サーバー環境にて十分検索エンジンの機能を果たした。しかしながら、指定されたサーバーを検索する地域イントラネット的な機能は十分検証できたが、インターネットへのアクセスなどで十分な

性能を発揮できなかった。主に、サーバー管理上の問題に依存するようである。

地図との連携は、理論的に困難な部分があることもわかった。今回は、手作業の補助による方法をとったが、今後、より自動的な手法について研究を深めていく必要がある。

5) 技術的所見

実験実施にあたって、以下のようないくつかの技術的所見をえた。参考情報として述べたい。

・ jdk のインストールの際に、PATH が自動で設定されないので手動で設定する必要があった。

(例) PATH=C:\jdk1.1.7b\bin

・ Namazu のインストールの際に、デフォルトのフォルダ以外にインストールする場合には、PATH を手動で設定する必要があった。

(例) PATH=C:\namazu\bin

・ Namazu のインデックスファイルのデフォルト値は、C:\namazu\bar\namazu\index であった。

・ Namazu.cgi をブラウザで呼び出す際は、namazu.cgi.exe とする必要があった。

・ IP アドレスに対応する FQDN が設定できないため通信の設定ができなかった。

・ UNICGI は起動に必要なだが、これがなくてもトラベルしてきた Aglets を受けて渡すことが可能であった。

・ WinNT と Win2000での設定はほぼ同じであった。

4. 考察と今後の課題

本研究では、地域情報ネットワーク構築の基本的要素として福祉エージェントの意義とその可能性について実証実験に基づく考察を行ってきた。現時点では、モバイルエージェント技術を活用して分散サーバー環境での地域イントラネットへのアクセスに関しては、ほぼ当初の成果は得られた。しかしながら、地図情報との連携、外部インターネットサイトへのアクセスと

情報活用に関しては少なからぬ課題も発見した。

このような実証実験が、真に情報ネットワークが実務、あるいは地域に根づくためには、不可欠な過程であると思う。今後、現場への適用を通じて課題のさらなる展開をすすめていきたい。

参考文献

- [1] 『地域福祉情報ネットワーク構築を目指して、共同研究プロジェクトー中間報告書』岐阜経済大学、2000年。
- [2] 『地理情報システムの取組』財団法人岐阜県市町村行政情報センター、2000年。<http://www.aic.pref.gifu.jp/kohosi/no89/4.html>
- [3] 相良、有川、坂内：ジオリファレンス情報を用いた空間情報媒介システム、情報研報 (DB研)、2000年 (<http://www.csis.u-tokyo.ac.jp/~sagara/research/currentresearch.html>)