モバイルエージェントによる Web ページ構築フレームワーク *MiPage* の提案

A Mobile Agent-based Web Page Constructing Framework *MiPage*

福田 直樹
Naoki FUKUTA

名古屋工業大学知能情報システム学科^{†1}
Department of Intelligence and Computer Science, Nagoya Institute of Technology
fukuta@ics.nitech.ac.jp, <http://www-toralab.ics.nitech.ac.jp/~fukuta/>

大園 忠親
Tadachika OZONO

(同 上)
chika@ics.nitech.ac.jp, <http://www-toralab.ics.nitech.ac.jp/~chika/>

新谷 虎松
Toramatsu SHINTANI

(同 上)
tora@ics.nitech.ac.jp, <http://www-toralab.ics.nitech.ac.jp/professor/tora.html>

keywords: mobile agent, world wide web, logic programming

Summary

In this paper, we present a programming framework, '*MiPage*', for realizing intelligent WWW applications based on the mobile agent technology. On the framework, an agent is programmed by using hyper text markup language and logic programming language. To realize the framework, we designed a new logic programming environment '*MiLog*', and an agent program compiler '*MiPage Compiler*'. The framework enables us to enhance both richness of the services and manageability of the application.

1. は じ め に

本論文では、Web ページに埋め込み可能で、かつ移動可能なエージェントを実現するためのフレームワーク *MiPage* を提案する。本フレームワークの特長は、1) Web ページを表現する HTML 文書へプログラムを直接埋め込むことが可能なため、表示結果の設計が容易となる点、2) プログラムの記述にモバイルエージェント記述言語 *MiLog*[Fukuta 01] を利用可能とすることで、強モビリティを利用したモバイルエージェントプログラムの記述が容易に行える点、および 3) HTTP のリダイレクト機構を利用することで、エージェントの移動を意識させることなくプログラムが実行可能である点の3つである。

Web ページにモバイルエージェントを埋め込むことを可能とすることにより、新しいスタイルの WWW アプリケーションが生まれる可能性がある。特に、複数サーバを利用するアプリケーションの記述が飛躍的に容易になる点が注目される。従来のサーバクライアント型のアプリケーション構築モデルでは、各ホストごとにサーバプログラムを用意し、それらのサーバプログラム間での通信プロトコルを規定して、サーバプログラム間で同期をとりながら動作させるように、プログラムを設計する

必要があった。本フレームワークでは、モバイルエージェントとしてアプリケーションを記述することにより、各ホストにはモバイルエージェントの実行環境のみを動作させ、アプリケーション固有のサーバプログラムの記述、通信プロトコルの規定、およびサーバプログラム間での同期処理を省略することを可能とする。

想定されるアプリケーションには、分散型グループ意思決定支援システム、情報収集システム、および情報モニタリングシステムがある。

分散型グループ意思決定支援システムでは、各ユーザごとにモバイルエージェントが割り当てられる。エージェントはユーザと WWW ブラウザにより対話する。ユーザと対話する場面では、ユーザの手元にエージェントを移動させ、ユーザとの対話の応答速度を向上させる。一方で、他のエージェントとの密な通信が必要な場合（交渉を行う場合など）は、エージェントを通信相手の近くに移動させることで、利用ネットワーク帯域の低減を行う。具体的な例としては、分散会議スケジューリングシステム [Shintani 01] などが考えられる。

情報収集システムでは、エージェントは、情報収集時にネットワークに常時接続された環境に移動して、ユーザと情報源との間の通信帯域に制限されることなく情報収集を行う。情報収集を行うエージェントは *MiPage* を用いて

^{†1} 現在、静岡大学情報学部

作成され、ユーザはエージェントにブラウザを介してアクセスすることにより情報収集の途中経過および結果を閲覧することが可能となる。同種のシステムに Mobeet[Mob] がある。

情報モニタリングシステムでは、エージェントはネットワーク上の特定の情報の変化を監視しており、情報が特定のパターンで変化した場合に、ユーザに電子メールを送付するなどの行動を実行する。エージェントが *MiPage* で構成されることにより動作中のエージェントの状態を WWW ブラウザを介して閲覧することが可能となる。エージェントがモバイルエージェントとして構成されることで、システムの負荷およびネットワーク速度等から、適切なホストをエージェント自身が選択して情報の監視を行うアプリケーションの設計が容易となる。具体的な例としては、オンラインオークションにおける商品価格の変化に対応して入札を代行するシステムとして、*BiddingBot* システム [Ito 00] がある。

本論文の構成は、次のようになっている。2 章では、*MiPage* フレームワークの概観を示し、システムの設計を明らかにする。3 章では、*MiPage* フレームワークを用いて実際にモバイルエージェントを構築する方法を述べる。4 章では、関連研究および関連技術との比較を行い、*MiPage* フレームワークの新規性を明らかにする。5 章で、本研究において今後解決されるべき課題について議論する。6 章で、本研究の成果をまとめる。

2. *MiPage* フレームワークの構成

2.1 システムの構成

システムの構成図を図 1 に示す。本システムは、論理プログラムを含んだ HTML 文書を *MiLog* プログラムに変換する *MiPage* コンパイラ、*MiLog* プログラムから *MiLog* エージェントを生成しそれを実行するための *MiLog* 実行環境、および HTTP リクエストに対応するエージェントに受け渡す HTTP リクエストルータの 3 つから構成される。エージェント設計者は、論理プログラムを含んだ HTML 文書 (*MiPage* プログラムと呼ぶ) を記述する。エージェントが利用者によって利用される場合、そのエージェントがまだ実行環境上に存在しない場合には、*MiPage* コンパイラによって *MiLog* プログラムが生成され、その *MiLog* プログラムが新たに生成されたエージェントにロードされる。すでにそのエージェントが存在する場合には、HTTP リクエストルータを経由してそのエージェントに利用者のブラウザから送られた HTTP リクエストが渡される。エージェントによる HTTP リクエストの処理結果は HTTP リクエストルータに戻され、HTTP リクエストルータから利用者のブラウザに処理結果が HTML 文書として渡される。

図 2 *MiPage* プログラムにおける HTML 文書へのカウンタ機能の埋め込み

2.2 *MiPage* プログラムと *MiPage* コンパイラ

MiPage プログラムは、HTML 文書中に特定のタグを埋め込むことにより記述される。図 2 に *MiPage* プログラムの例を示す。図 2 のプログラムでは、アクセスされるごとに値を 1 ずつカウントアップする機能を持つ Web ページを記述している。*MiPage* プログラムでは、*<HEAD>* タグ中の *<SCRIPT>* タグ中にプログラムを格納する (図 2 の (a) の部分)。*<SCRIPT>* タグ中のコードが *MiLog* プログラムであることを明示するために、*<SCRIPT>* タグのオプションとして *LANGUAGE="MiLog"* を指定する。例えば、図 2 では、呼び出されるごとに値を 1 増加させる述語 *currentValue/1* を定義している。述語 *currentValue/1* では、初期値を節データベースから取り出し、その値に 1 を加算した後、加算後の値を節データベースに再格納している。*<SCRIPT>* タグ内に記述したプログラムは、コメントタグを変形した特殊なタグ中に記述された問い合わせによって起動される。例えば、図 2 のプログラムでは、*<!--?-currentValue(X).-->* (図 2 の (b) の部分) が問い合わせの記述である。問い合わせ *?-currentValue(X).* が実行され、変数 *X* への束縛値が問い合わせ記述部分と置き換わる。

図 3 に、図 2 のプログラムの *MiPage* コンパイラによるコンパイル結果を示す。コンパイル結果として出力される *MiLog* コードは、*MiPage* プログラム中に記述された論理プログラム部分 (図 3 の (a) の部分)、HTTP リクエストハンドラの定義 (図 3 の (b) の部分)、および HTTP リクエストハンドラ中の動作 (図 3 の (c) の部分) の 3 つの部分から構成される。*MiLog* エージェントに対する HTTP リクエストは、HTTP リクエストルータによって問い合わせ *doService/2* に変換される。問い合わせ *doService/2* では、第 1 引数に、その問い合わせの識別子が入り、第 2 引数にエージェントによるリクエストの処理結果がリスト構造で束縛される。第 2 引数への束縛結果は HTTP リクエストルータによって HTML 文書

図 1 システムの構成

に変換され、HTTP レスポンスとして利用者の WWW ブラウザへ返される。

図 3 MiPage コンパイラによる出力

図 4 に図 2 のプログラムの実行例を示す。WWW ブラウザからのアクセスがあるたびに、カウンタの表示が 1 ずつ増加する。

図 4 WWW ブラウザ上での表示結果

2.3 HTTP リクエストルータ

HTTP リクエストルータは *MiLog* 実行環境内部に含まれ、WWW ブラウザ等からの HTTP リクエストを *MiLog* エージェントに受け渡す役割を果たしている。HTTP リクエストの例を図 5 に示す。HTTP リクエストでは、WWW サーバに対する要求を、URL、要求のオプションパラメータ、およびその URL の扱い方（メソッド）として記述する。たとえば、WWW ブラウザのアドレスバーから URL として `http://127.0.0.1:17008/agent?item=ruby&action=delete` を入力した場合、IP アドレス 127.0.0.1 上でポート番号 17008 で動作している WWW サーバに対して図 5 の要求が送信される。図 5 の要求では、`agent?item=ruby &action=delete` が URL として指定されている。URL 中の ‘?’ 記号より後の部分はこの要求に対するパラメータを意味し、ここでは、項目 ‘item’ および ‘action’ の値がそれぞれ ‘ruby’、‘delete’ であることを示している。図 5 の要求では、この URL に対する処理の方法として、‘GET’ が指定されている。要求のオプションパラメータには、WWW ブラウザの種類、および WWW ブラウザが受け取ることができるデータ型についての定義が含まれている。

```
GET agent?item=ruby&action=delete HTTP/1.1
Accept: */*
Accept-Language: ja,en;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Host: 127.0.0.1:17008
Connection: Keep-Alive
```

図 5 HTTP リクエストの例

HTTP リクエストルータは HTTP リクエストを受け取ると、そのリクエストを URL、要求のオプションパラメータおよびその URL の扱い方に分離する。URL は、さらにそれをファイルパスおよびパラメータ列に分解する。HTTP リクエストルータは、ファイルパスをエージェント名とみなして、該当するエージェントがその *MiLog*

実行環境上に存在するかどうかを調べる。もし、該当するエージェントが存在すれば、そのエージェントに対して、HTTP リクエストの処理を問い合わせる。この問い合わせでは、まず、HTTP リクエスト中のオプションパラメータが `option( ID, NAME, VALUE)` の形式で対象とするエージェントの持つ節データベース中に追加 (`assert`) される。次に、HTTP リクエストに対する処理を求める問い合わせ `doService/2` が対象とするエージェント上で起動される。問い合わせ `doService/2` の実行終了後、追加 (`assert`) されていたオプションパラメータが削除 (`retract`) され、`doService/2` の実行結果が HTTP レスポンスとして整形され、HTTP リクエスト元へ返信される。

URL で複数のオプションパラメータが指定された場合、それぞれのオプションパラメータが個別に節データベースに追加される。例えば、

```
http://133.68.221.183:17008/agent
?opt1=val1&opt2=val2&opt3=val3
```

という URL では、3つのオプションパラメータ (`opt1 = val1`, `opt2 = val2`, `opt3 = val3`) が指定されている。

この場合、それぞれのオプションパラメータに対して、

```
option( agent, opt1, val1).
option( agent, opt2, val2).
option( agent, opt3, val3).
```

という3つの節がデータベースに追加される。ここで、`agent` はこの HTTP リクエストに割り当てられた識別子を意味し、HTTP リクエストごとに値が変化する場合がある。これらの各オプションパラメータへのアクセスは、

```
?- option( ID, opt1, VALUE).
```

のように、パラメータ名をキーとして問い合わせを実行することで行う。この問い合わせでは、変数 `VALUE` の束縛として `val1` を得ることができる。

図6に、HTTP レスポンスの例を示す。HTTP レスポンスは、HTTP リクエストに対する処理の結果、レスポンスのオプションパラメータ、およびレスポンス本体から構成される。図6の HTTP レスポンスは、図3のプログラムによって生成された *MiLog* エージェントからの応答の例である。

```
HTTP/1.0 200 OK
Content-Type: text/html

<html><head></head><body>
Current value is 1
</body></html>
```

図6 HTTP レスポンスの例

3. マイグレーションの記述

本節では、*MiPage* プログラム中でのマイグレーションの記述とその実現方法を述べる。*MiLog* エージェントは、HTTP リクエスト処理中に他の *MiLog* 実行環境上へ移動することが可能である。ここで課題となるのは、HTTP リクエストの処理結果をどのようにブラウザに返すのか、および移動後のエージェントにブラウザからどのようにアクセスを行うかである。本節では特に、上記2点の実現方法について詳細に述べる。

図7に、モビリティを用いた *MiPage* プログラムの例を示す。図7の例では、特定のハイパーリンクをブラウザ上でクリックすることにより、その WWW アプリケーションを特定の計算機上へ移動させることができる。

図7のプログラムを実行した場合における処理の過程を示す。まず、ブラウザからの HTTP リクエストを HTTP リクエストルータが受け取る。HTTP リクエストルータは HTTP リクエストを解析し、エージェント間通信を用いて対象とするエージェントに HTTP リクエストの処理を依頼する。処理を依頼されたエージェントでは、Web ページをレンダリングするために、問い合わせ `?-currentLocation(X)`、および `?- whereToGo(X)` が実行される。問い合わせ `?-currentLocation(X)` の実行時における `move/1` 述語の実行により、エージェントは他の実行環境へ移動する。このとき、移動を行ったことがメタエージェント上で記録される。HTTP リクエストの問い合わせに対する応答は、メタエージェントによって移動先の実行環境から HTTP リクエストを受け取った実行環境へ送られ、WWW ブラウザ上に結果が表示される。移動を行ったエージェントに対するアクセスがあった場合には、HTTP のリダイレクト機能を利用してブラウザの接続先をエージェントの移動先ホストに切り替える。なお、図7プログラム中の述語 `thisHost/1` および `concat/2` はそれぞれ、現在いるホストのアドレスを得る、および複数のアトムをつなげて1つの長いアトムを生成する機能を持つ *MiLog* 固有の組み込み述語である。

図8にリダイレクトを行うための HTTP レスポンスの例を示す。ブラウザは、図8に示されるような HTTP レスポンスを受け取ると、自動的にリダイレクト先の URL へ HTTP リクエストを再発行する。リダイレクト先の URL は、HTTP ヘッダフィールド中の 'Location' によって指定される。HTTP リダイレクト機能により、アプリケーションのユーザはエージェントの移動を意識することなくアプリケーションを利用することが可能となっている。

4. 関連研究

本章では、関連研究および関連技術を示し、本フレームワークにおける新規性を明らかにする。

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="MiLog">
    currentLocation(X) :-
      option(_,move,LOCATION),
      location,LOCATION,HOSTADR),
      move(HOSTADR),
      thisHost(X),!.
    currentLocation(X) :-
      thisHost(X).

    whereToGo(X) :-
      myname(MYNAME),
      concat(
        ['<A HREF="',MYNAME,
          '?move=1">Go to location 1</A><BR>',
          '<A HREF="',MYNAME,
          '?move=2">Go to location 2</A><BR>',
          '<A HREF="',MYNAME,
          '?move=3">Go to location 3</A><BR>'],
        X).

    location(1,'http://133.68.221.151:17008/').
    location(2,'http://133.68.221.152:17008/').
    location(3,'http://133.68.221.153:17008/').
  </SCRIPT>
</HEAD>
<BODY>
  Current location is
  <!--?-currentLocation(X).-->
  <BR>
  <!--?-whereToGo(X).-->
</BODY>
</HTML>

```

図 7 モビリティを用いた *MiPage* プログラムの例

```

HTTP/1.0 302 Found
Content-Type: text/html
Location: http://www-toralab.ics.nitech.ac.jp/

```

図 8 リダイレクトを要求する HTTP レスポンス

・静的なアプリケーション

WWW 上のサービスをモバイルエージェントではなく静的なアプリケーションで構築するための実装方法として、CGI に基づく実装方法、PHP[Bakken],JSP[Sun 01],ASP[Microsoft] 等のページ記述言語を用いる方法がある。*MiPage* では、Web ページにモバイルエージェントを埋め込み可能とすることにより、複数のホストを利用した Web アプリケーションの構築を容易としている点が異なる。

・モバイルエージェントの実装技術

これまでに、Telescript[White 94] や Aglets[Lange 98] など多くのモバイルエージェントシステムが構築されている。これらのモバイルエージェントシステムの研究成果から、局所的なリソースへのローカルアクセスによる処理の効率化[Brewington 99]、および既存のアプリケーションの相互接続への適用[川村 99]、が示されている。しかし、これらの研究では WWW とモバイルエージェン

トの統合については議論されていない。本フレームワークでは、WWW へのモバイルエージェントの埋め込みについて議論を行っている点が異なる。

・論理プログラミングに基づく WWW の再構築

Loke らは、文献 [Loke 98] で論理プログラミングに基づく WWW の再構築について述べている。Loke らは、ハイパーリンク構造などの Web ページ上の情報が論理プログラミングによって宣言的に表現され、WWW 上の情報をソフトウェアにとって扱いやすいものとする事で、WWW 上の情報を効果的に活用する可能性を示している。本研究ではアプリケーションロジックの記述のために Web ページ中に論理プログラミングを含ませることができているが、本研究ではアプリケーションを構築するための手法を実現することに重点を置いている点が異なる。

5. 議 論

5.1 エージェントの移動とセキュリティ

現状の *MiPage* の実装では、エージェントは、移動先と移動元との認証が成功した場合にのみ、移動可能となっている。これは、先行研究である *MiLog* と同様のセキュリティモデルを用いていることによる。*MiLog* では、予め信頼されたホスト同士が共通鍵を持ち、エージェント移動時に移動先のホストと認証を行っている。移動を拒絶する権利は、移動元と移動先の両者が持っており、両者がともに移動を許可した場合にのみエージェントは移動する。エージェント自身は、それを実行するホストからの悪意ある改ざんに対抗するための手段をもっておらず、またホスト側もエージェントの振る舞いに対して制限を加えない。これらのセキュリティ指針は、必ずしも最適なものではないが、プロトタイプアプリケーション作成時に、セキュリティ関連の配慮の負担を最小限とし、アプリケーションの開発の行いやすさを向上させる場合に適していると考えられる。構造的には、*MiLog* エージェントはインタプリタ上で動作するため、サンドボックスモデルに基づく動作の制限は容易であり、移動先環境側でのエージェントの悪意ある行動を制限することは可能である。エージェント自身のセキュリティについては、通信路上での盗聴を防ぐことは可能であるが、悪意あるホストからの盗聴や改ざんを防ぐための手立ては実現されておらず、今後の課題となる。

5.2 エージェントの名前管理とエージェント間通信

エージェントの名前は、エージェント自身の名前とそのエージェントが生成されたホストの名前の組み合わせによって表現される。たとえば、ホストの名前を '133.68.221.183:17008'、エージェント自身の名前を 'agent' とすると、エージェントの識別名は 'agent@133.68.221.183:17008' となる。*MiPage* では、各ホストに一意的

名前を割り当て、かつ各ホスト上で生成されるエージェント名が一意となるようにすることにより、エージェント名の衝突を避けている。

エージェント間の通信では、*MiLog* で用意されたエージェント間問い合わせ述語、あるいは HTTP アクセスによる通信の 2 つを用いることが可能となっている。*MiLog* では、HTTP サーバ機能のみでなく、HTTP クライアント機能を容易に実現するための組み込み述語を多数用意しており、*MiPage* の提供する HTTP のリダイレクト機能によるスイッチが利用可能となっている。

エージェント間問い合わせ述語では、エージェントの移動履歴に基づいて、問い合わせ先の探索および問い合わせ結果の返信を行う機構が *MiLog* 上において用意されるが、これらの通信はベストエフォート型であり、必ず通信が成功することを保証するものではない。効果的なエージェント間通信、および通信の保証は今後の課題である。

6. お わ り に

本論文では、Web ページに埋め込み可能で、かつ移動可能なエージェントを実現するためのフレームワーク *MiPage* を提案した。*MiPage* では、*MiPage* コンパイラを実装することにより、Web ページを表現する HTML 文書へのプログラムの埋め込みを可能とした。*MiPage* では、*MiLog* エージェントフレームワークの拡張として実装することにより、*MiLog* エージェント記述言語をエージェントプログラムの記述に利用可能とした。*MiPage* では、HTTP のリダイレクト機構を利用することにより、エージェントの移動を意識せずにエージェントプログラムの記述を行うことを可能とした。関連研究との差分を示し、今後の課題として、セキュリティ、およびエージェントの名前管理とエージェント間通信の 2 点から考察を行った。

謝 辞

本研究を進めるにあたり、北陸先端科学技術大学院大学知識科学教育研究センターの伊藤孝行助教授から、数々の有用な助言を頂いた。本論文の査読者の方々より、本論文を洗練する上での多くの有用な指摘および助言を頂いた。ここに謝意を表する。

◇ 参 考 文 献 ◇

- [Bakken] Bakken, S. S., Aulbach, A., Schmid, E., Winsted, J., Wilson, L. T., Lerdorf, R., Suraski, Z., Zmievski, A., and Ahto, J.: PHP Manual: (<http://www.php.net/manual/en/>).
- [Brewington 99] Brewington, B., Gray, R., Moizumi, K., Kotz, D., Cybenko, G., and Rus, D.: Mobile Agents for Distributed Information Retrieval, in Klusch, M. ed., *Intelligent Information Agents*, chapter 15, pp. 355–395, Springer-Verlag (1999).

- [Fukuta 01] Fukuta, N., Ito, T., and Shintani, T.: A Logic-based Framework for Mobile Intelligent Information Agents, in *Poster Proc. of the Tenth International World Wide Web Conference(WWW10)*, pp. 58–59 (2001).
- [Ito 00] Ito, T., Fukuta, N., Shintani, T., and Sycara, K.: BiddingBot: A Multiagent Support System for Cooperative Bidding in Multiple Auctions, in *Proc. of 4th International Conference on Multi Agent Systems (ICMAS'2000)*, pp. 435–436 (2000).
- [川村 99] 川村, 田原, 長谷川, 大須賀, 本位田: Bee-gent: 移動型仲介エージェントによる既存システムの柔軟な活用を目的としたマルチエージェントフレームワーク, *電子情報通信学会論文誌 (D-I)*, Vol. J82-D-I, No. 9, pp. 1165–1180 (1999).
- [Lange 98] Lange, D. and Oshima, M.: *Programming and deploying Java mobile agents with Aglets*, Addison-Wesley (1998).
- [Loke 98] Loke, S. W. and Davidson, A.: LogicWeb: Enhancing the Web with logic programming, *The Journal of Logic Programming*, Vol. 36, pp. 195–240 (1998).
- [Microsoft] Microsoft, : Active Server Pages: (<http://msdn.microsoft.com/asp/>).
- [Mob] Mobeet(<http://mobeet.ex.nii.ac.jp/>).
- [Shintani 01] Shintani, T. and Ito, T.: Cooperative Meeting Scheduling among Agents base on Multiple Negotiations, in *Proc. of 6th International Conference on Cooperative Information Systems(CoopIS2001)* (2001).
- [Sun 01] Sun, : Java Server Pages Quickstart Guide (2001), (<http://java.sun.com/products/jsp/docs.html>).
- [White 94] White, J. E.: Mobile agents make a network an open platform for third-party developers, *IEEE Computer*, Vol. 27, No. 11, pp. 89–90 (1994).

〔担当委員：武田英明〕

2001 年 9 月 17 日 受理

著 者 紹 介

福田 直樹 (正会員)

1997 年名古屋工業大学工学部知能情報システム学科卒業。2002 年同大学大学院工学研究科電気情報工学専攻博士後期課程修了。同年より、静岡大学情報学部情報科学科助手。現在に至る。博士 (工学)。論理プログラミングと知的モバイルエージェントに興味を持つ。ACM, 電子情報通信学会, 日本ソフトウェア科学会, 情報処理学会各会員。

大園 忠頼 (正会員)

1995 年名古屋工業大学工学部知能情報システム学科卒業。2000 年同大学大学院工学研究科電気情報工学専攻博士後期課程修了。同年より同大学工学部知能情報システム学科助手。現在に至る。博士 (工学)。情報エージェントの研究に従事。AAAI, ACM, 情報処理学会, 日本ソフトウェア科学会, 各正会員。

新谷 虎松 (正会員)

1982 年東京理科大学大学院修士課程修了。同年富士通 (株) 国際情報社会科学研究所入所。知識システム, 論理プログラミングなどの研究に従事。1993 年名古屋工業大学知能情報システム学科助教授。1999 年同大学同学科教授。1999 年から 2000 年にかけて米国カーネギーメロン大学ロボティクス研究所客員研究員。現在に至る。博士 (工学)。意思決定支援システム, マルチエージェントシステム, 電子商取引システム, WWW 情報検索の研究に従事。AAAI, 電子情報通信学会, 情報処理学会, 日本ソフトウェア科学会各会員。