

# WWW 情報統合のためのマルチキャラクタ インタフェースとその評価

北村 泰彦 山田 晃弘 山本 太三 辻本 秀樹 辰巳 昭治

本論文では WWW 情報統合プラットフォームの一つとして、ユーザが複数のアニメーションキャラクタとインタラクションしながら協調的に情報検索や情報統合を行うマルチキャラクタインタフェース (MCI: Multiple Characters Interface) を提案する。MCI ではユーザはキャラクタインタフェースをもつ複数の情報エージェントをプラットフォーム上でプラグアンドプレイ的に協調させることが可能である。ユーザに情報統合の過程を隠蔽する従来の情報統合システムとは異なり、MCI では情報統合の過程がユーザに対してオープンであり、ユーザはキャラクタのインタラクションを通してその過程を理解し、それを変更することができる。MCI に基づく情報統合プロトタイプとして、3 体のキャラクタによる協調型レシピ検索システム Venus & Mars を開発した。さらに、その評価を Wizard of Oz 法を用いて行った。その結果、複数のキャラクタはユーザの検索要求の範囲を広げる効果

もあることが明らかになった。

## 1 はじめに

WWW(World Wide Web) は 2 つの観点からインターネット上のオープンな情報システムであると見なすことができる。1 つ目は情報提供者に対してオープンである点である。情報提供者はインターネットにつながったコンピュータ上で Web サーバを起動すれば、即座に情報を世界に発信することができる。このような容易さのゆえに、インターネット上の Web サイトの数は日々急速に増加している。しかし、その一方では関連のある情報が多数の Web サイト間に散在する傾向にあり、情報検索や情報統合タスクを困難にしている。その克服のために様々な検索エンジン、情報統合システム[7][21]、情報エージェント[11]が開発されている。

情報統合システムは、そのシステムを構成するそれぞれの情報源に付加価値を与える。例えば、レシピページを蓄積し、料理の食材を入力することで検索できる Web サイトがある<sup>†1</sup>。また、健康と食べ物についての情報を蓄積した Web サイトもある<sup>†2</sup>。通常はそれぞれの Web サイトは独立した情報源として動作するが、もしもこれらのサイトの情報を統合できれば、「風邪に効く料理のレシピを教えてください」というような、1 つの Web サイトだけでは答えることができない複合的な質問に答えることができるように

---

Multiple-Character Interface for WWW Information Integration and Its Evaluation.

Yasuhiko Kitamura, 大阪市立大学大学院 工学研究科 情報工学専攻, Department of Information and Communication Engineering, Graduate School of Engineering, Osaka City University.

Teruhiro Yamada, 三洋電機株式会社, SANYO Electric. Taizo Yamamoto, NTT ブロードバンドイニシアティブ株式会社, NTT Broadband Initiative.

Hideki Tsujimoto, 日立ソフトウェアエンジニアリング株式会社, Hitachi Software Engineering.

Shoji Tatsumi, 大阪市立大学大学院 工学研究科 情報工学専攻, Department of Information and Communication Engineering, Graduate School of Engineering, Osaka City University.

---

†1 例: <http://www.kikkoman.co.jp/homecook/>

†2 例: <http://www.shokuiku.co.jp/index.html>

なる。

WWW のもう一つの特徴はユーザに対してオープンである点である。インターネットは我々の社会に急速に浸透し、日々の生活を支える社会基盤の 1 つとなっている。そのユーザ層は、特別な訓練や教育を受けたコンピュータの専門家から、子供やお年寄りのような初心者にまで広がっている。現在、Web サイトへのアクセスには Internet Explorer や Netscape Navigator のような Web ブラウザが最も広く使われているが、将来は初心者ユーザでも簡単に操作できる、よりユーザフレンドリーなツールやインタフェースが期待される。この実現に向けては、生き物のようなアニメーションキャラクタを用いた様々なキャラクタエージェントが開発され<sup>[3]</sup>、商用目的のキャラクタエージェントとしては Extempo<sup>†3</sup>、Haptek<sup>†4</sup>、Virtual Personalities<sup>†5</sup>、Artificial Life<sup>†6</sup> のようにインターネット上で利用可能なものもある。これらのエージェントは英語などの自然言語でユーザと会話し、身振りを伴って Web サイトの案内をしてくれる。

これまで、オープンな Web 情報システムに対する上記の 2 つのアプローチはそれぞれ個々の研究課題として論じられていた。それに対して、本論文ではこれらのアプローチを一つのプラットフォームに統合し、個別のキャラクタインタフェースを持った複数のエージェントがインタラクションを行うマルチキャラクタインタフェース (MCI: Multiple Characters Interface) を提案する。

MCI の利点は以下のように要約される。

- ユーザと情報源の間に親しみやすいインタフェースを提供する。3.1 節でも示すように、ユーザはエージェントにキーワードを含んだ自然言語で命令することができ、情報源ごとの複雑なコマンドを学習する必要がない。アニメーションキャラクタは親しみやすく、ユーザを楽しませる能力を秘めている。
- エージェントは情報検索や情報統合においてユー

ザを支援するために協調する。3.2 節でも示すように、エージェントは他のエージェントが必要としているキーワードを自発的に発話することで、複数のエージェントによる協調的な情報検索が可能になる。

- 3.2 節で示すように、ユーザはキャラクタを介してタスク実行中にエージェントのふるまいを修正できる。また、ユーザは好みのエージェントを選ぶことで情報統合のためのエージェントの組み合わせを柔軟に変更することができる。
- 3.2 節でも示すように、情報検索や情報統合の過程をキャラクタ間の会話を通してユーザに示すことで、ユーザはシステム内で起きていることを直感的に理解できる。これは、4 章でも述べるように、情報検索に関してユーザの興味の範囲を広げる効果もある。

本論文では、2 章において MCI のアーキテクチャとその実装について述べる。特に、情報源がインターネット上に分散しながらも、複数のキャラクタがクライアント上でプラグアンドプレイ的にインタラクションする仕組みについて議論する。3 章では MCI を用いた情報統合プロトタイプとして、レシピページを検索するために 3 体のキャラクタが協調する、Venus & Mars (Virtual Environment for Novice User Support and Mult-Agent Recommendation System) システムについて議論する。4 章では Wizard of Oz 法を用いたその評価について述べる。Venus&Mars はプロトタイプの段階であり、ユーザとの対話能力が十分ではない。そこでウィザードと呼ばれる人間がエージェントの代わりにする Wizard of Oz 法を用い、MCI の持つ能力を明らかにしようとした。ここではキャラクタの数、キャラクタ同士の協調、キャラクタの役割分担がユーザの情報検索にどのような影響を及ぼすかを明らかにした。最後に 5 章では関連研究について述べ、6 章でまとめと今後の課題を述べる。

## 2 マルチキャラクタインタフェース

### 2.1 アーキテクチャ

MCI では図 1 に示すように、検索エンジン (Search Engine) やデータベース (Database) などのインター

†3 <http://www.extempo.com>

†4 <http://www.haptek.com>

†5 <http://www.vperson.com>

†6 <http://www.artificial-life.com>

ネット上の情報源が、エージェントの相互作用を介して統合され、ユーザに提供される。一つのエージェントは本体部（情報エンジン）と頭部（キャラクタ）から構成される。情報エンジン (Information Engine) はインターネット上で分散した個別のサーバに存在し、検索エンジンやデータベースから特定領域情報の検索や収集を行う。エージェントの中にはユーザの個人情報や管理し、情報検索タスクを支援するような個人エージェントも加えることも可能である。

エージェントの頭部は MS-Agent<sup>†7</sup> などのアニメーションキャラクタインタフェースによって具現化される。ユーザはクリックしてメッセージを与えることによってキャラクタにアクセスでき、キャラクタは身振りを伴って話すことでユーザに反応することができる。

エージェントはユーザの行動をセンサー (Sensor) を通じて検知し、そのセンサーデータはネットワークを介して情報エンジンに送られる。センサーデータはセンサーコントローラ (Sensor Controller) を介して一旦センサーバッファ (Sensor Buffer) に蓄積される。情報エンジン内では Q インタプリタ (Interpreter) がセンサバッファのセンサーデータを読み出し、スクリプトにしたがってアクチュエータコマンドが生成される。アクチュエータコマンドはアクチュエータコントローラ (Actuator Controller) とネットワークを介してキャラクタに送られる。キャラクタはそのアクチュエータコマンドに従い動作する。

エージェントの動作は MCI 上の他のエージェントにも検知される。これはアクチュエータコントローラに送られたアクチュエータ命令がコミュニケーションサーバー (Communication Server) を介して他のエージェントのセンサーコントローラに送ることにより実現される。これにより複数のエージェントが関わる協調的な動作を実現することができる。

## 2.2 コマンドとシナリオ

センサーデータに対するエージェントのふるまいは、京都大学で開発されたインタラクション記述言語

Q[9] で記述されたシナリオによって決定される。表 1 にシナリオに記述可能な主要なセンサーコマンドとアクチュエータコマンドを示している。エージェントの機能に応じて他のオプションコマンドも追加することも可能である。

図 2 は Q 言語によって記述されたエージェントのシナリオ例である。シナリオは 1 つ以上のシーンから構成され、1 つのシーンは 1 つ以上のルールから構成される。ルールは 1 つのセンサーコマンドと 1 つ以上のアクチュエータコマンドから構成される。ここでは最初のシーン (シーン 1) でエージェントが現れ、シーン 2 へ移る。シーン 2 ではユーザや他のエージェントからの "Hello!" という発話が聞こえた場合には "Hello!" と発話する。またクリックされた場合には "I am itchy." と発話し、シーン 3 へ移る。シーン 3 ではクリックに対して "Please stop it!" と発話するようになる。一般的にはこのようなシナリオは状態遷移グラフとして記述することが可能である。レシビ検索に用いられた状態遷移グラフの具体例は図 5 と図 6 に示されている。

## 2.3 MCI の実装

MCI ではエージェントが起動されると、キャラクタはブラウザ上にロードされ、プラグアンドプレイでユーザや他のエージェントとインタラクションを開始する。MCI ではこの機能を実現するために HTML において画面分割の目的で用いられているフレーム機能を用いている。すなわち、MCI は図 3 に示すように一つの制御フレーム (Control Frame) と複数のエージェントフレーム (Agent Frame) を用いることで実装されている。

MCI が起動されると、制御フレームに Agent Manager, User Manager, Dialogue Manager がロードされる。Agent Manager はエージェントが起動されたり停止されたときにエージェントフレームを管理する。Agent Manager はエージェント起動要求に対して、指定されたエージェントサーバから Character Controller, Command Receiver, Command Transmitter のモジュールを、割り当てられたエージェントフレームにダウンロードする。Character Controller

<sup>†7</sup> [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msagent/agentstartpage\\_7gdh.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msagent/agentstartpage_7gdh.asp)

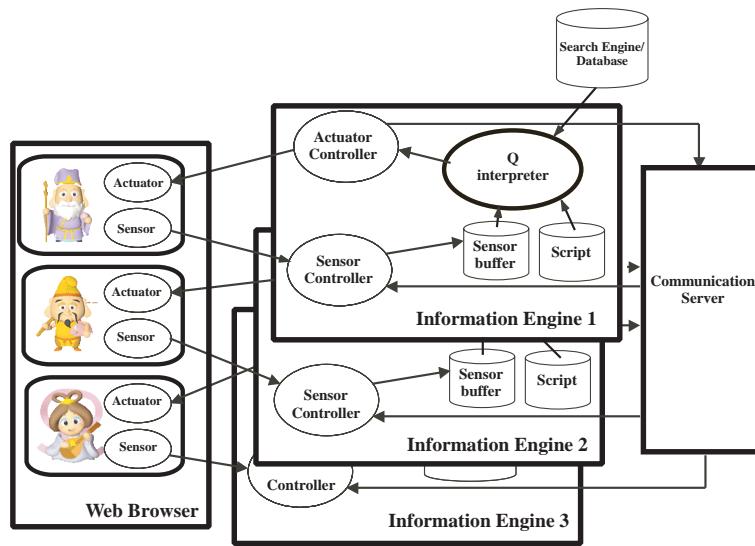


図 1 マルチキャラクタインタフェースのアーキテクチャ

表 1 主なセンサーコマンドとアクチュエータコマンド

センサーコマンド	
(?feel)	クリックされたかどうかを調べる
(?hear \$utterance [:from agent])	ユーザや他のエージェントの発話を聞く
(?find \$database_name :parameter parameter)	データベースから情報を取得する
アクチュエータコマンド	
(!show agent)	エージェントが現れる
(!hide agent)	エージェントが消える
(!speak utterance [:to agent])	エージェントが発話する
(!play_animation action)	エージェントがアニメーションを実行する
(!display_dialogue_box type [:title title])	ダイアログボックスを表示する
(!present url)	Web ページをブラウザに表示する

は JavaScript で記述されており、MS-Agent を使って実装されたキャラクタを制御する。これは図 1 の Sensor と Actuator に該当する。Command Receiver と Command Transmitter はキャラクタ（頭部）と情報エンジン（本体部）を接続する Java アプレットである。

User Manager は Cookie を用いることでユーザを

判別し、それによりエージェントはユーザによってそのふるまいを変えることができる。Dialogue Manager はエージェントの発話の順序を制御する。MCI ではそれぞれ独立な存在であるエージェントが自律的にふるまうことを仮定しており、制御なしには複数のエージェントが同時に発話することが起こりうる。そこで Dialogue Manager は一つのトークンを用いて

```

(scene1                                // Scene 1
  ((otherwise)
    (!show)                             // The agent appears
    (go scene2)))                       // and goes to Scene 2.
(scene2                                // Scene 2
  ((?hear "Hello!")                    // If the agent hears "Hello,"
    (!speak "Hello!")                  // he says "Hello."
    (?feel)                             // If the agent is clicked,
    (!speak "I am itchy.")              // he says "I am itchy."
    (go scene3)))                       // and goes to Scene 3.
(scene3                                // Scene 3
  ((?feel)                             // If the agent is clicked,
    (!speak "Please stop it.")          // he says "Please stop it."
    (go scene4)))                       // and goes to Scene 4.

```

図 2 Q 言語によるシナリオの例

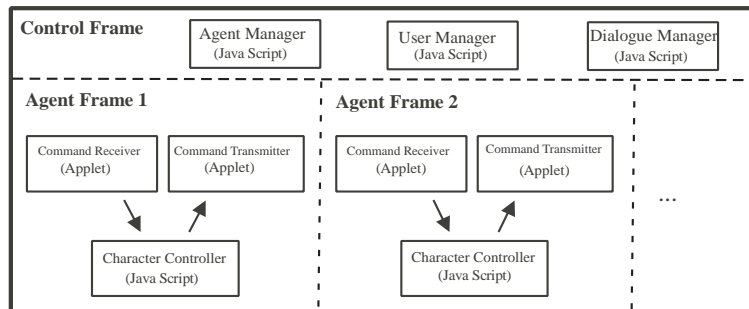


図 3 マルチキャラクタインタフェースの実装

発話制御を行う。エージェントが発話をする場合には Dialogue Manager にそのトークンを要求し、発話が終わるとそれを返却する。このような発話制御により複数のエージェントが同時に発話することを防いでいる。

MCI は以上のようなマルチエージェントアーキテクチャを採用しているので、タスク実行中のエージェントの追加や削除も可能である。これにより、多様なエージェントの組み合わせによる様々な情報検索や情報統合を可能にする。

### 3 Venus & Mars: 協調型レシピ検索システム

#### テム

Venus & Mars は複数のエージェントがレシピページを検索するために協調する協調型レシピ検索システムである。図 4 に示すように Internet Explorer 上に個人エージェント、レシピエージェント、健康エージェントを表す 3 体のキャラクタが登場し、検索結果が 2 つのフレームに表示される。左のフレームには検索結果のレシピページの URL 一覧が表示され、右のフレームには一覧からクリックされたページが表示される。デフォルトとして一覧の最初のページが右のフレームに表示される。

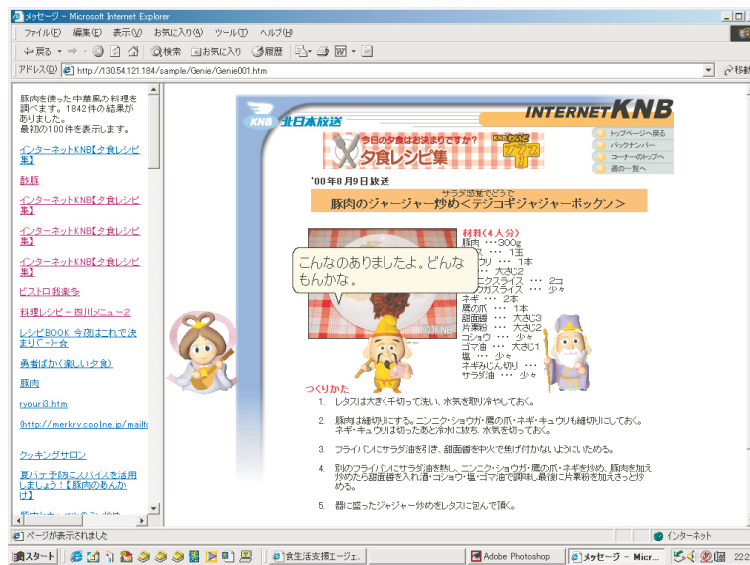


図4 Venus & Mars の実行画面

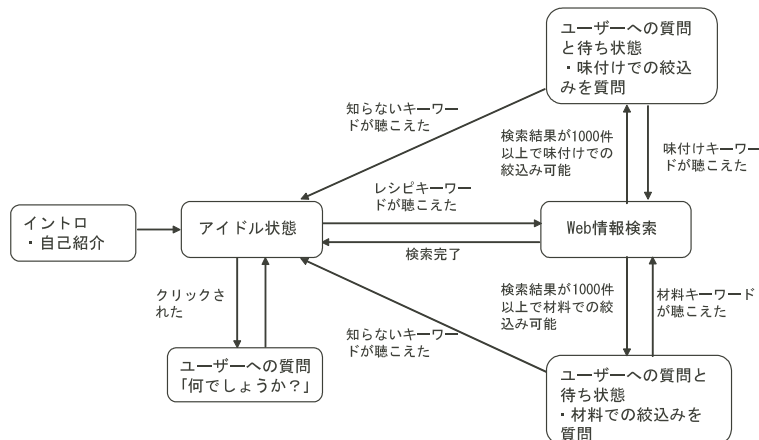


図5 レシピエージェントの状態遷移グラフ

### 3.1 エージェントとその機能

#### 3.1.1 レシピエージェント

レシピエージェントは図5に示される状態遷移グラフに従い、起動されると自己紹介に関する発話を行いアイドル状態に遷移する。アイドル状態においてエージェントは「豚肉を使った料理が食べたい」というようなレシピについてのキーワードを含んだ日本語の発話を認識する。エージェントの自然言語理解はキーワードのパターン照合に基づく単純な手法

である。レシピエージェントはレシピ、味付け、材料に関するキーワードを管理しており、発話中に関係するキーワードが含まれていれば状態遷移を行う。したがって現時点では「豚肉を使っていない料理が食べたい」と発話しても、豚肉を使った料理を検索してしまうという問題点がある。自然言語理解能力の高度化は今後の課題である。

レシピエージェントはアイドル状態においてレシピキーワード(例えば「豚肉」や「カレーライス」)を

含んだ発話に対し、そのキーワードを検索隠し味<sup>[15]</sup>とともに汎用検索エンジン<sup>†8</sup>で検索を行う。検索隠し味とは汎用検索エンジンの性能を向上させる領域特化キーワードである。Venus & Mars では「((材料 $\wedge$ 専門 $\wedge$ 商品) $\vee$ 大さじ)」<sup>†9</sup>という検索隠し味を用いることでレシピに関する Web ページを検索している。

レシピエージェントは検索エンジンからの検索結果を受け取り、それを Web ブラウザに表示する。もしその結果が 1000 件以上の場合には、自動的に味付け(洋風, 和風, 中華風など)あるいは材料についての質問を行い検索結果を絞り込む。どちらを選ぶかはランダムに選択されるが、一方の質問で絞り込みが十分でない場合はもう一方の質問を行う。ユーザからの返事を受け取ると、そのキーワードと検索隠し味を用いてもう一度検索を行う。

### 3.1.2 健康エージェント

健康エージェントはローカルデータベースに料理の食材と健康についての知識を持っている。健康エージェントの状態遷移グラフは省略するが、レシピエージェントよりも単純なものである。健康エージェントは食材と健康に関するキーワードを管理しており、料理の食材や健康に関連するキーワードを含んだ発話に対して、ローカルデータベースの中から「風邪には長ネギが良いんじゃないよ。」のように食材と健康の関係を示す 306 種類のコメントを発話することができる。

### 3.1.3 個人エージェント

個人エージェントはユーザの発話を監視しており、必要なときに発話履歴を参照してユーザの代わりに検索キーワードを提示する。個人エージェントの状態遷移グラフを図 6 に示す。

例えば、レシピエージェントがユーザの好みの味付けを尋ねてきたときにユーザが自分の好みの味付けについて発言すると、個人エージェントはその質問と答えの組をローカルデータベースに登録し、以後同じ質問がなされたときにはユーザの代わりに発話する。ユーザがその発話が気に入らない場合は正しい答え

を発話してもかまわない。その場合、個人エージェントはローカルデータベースを更新する。現在のところ個人エージェントが管理するのは質問と答えの組だけであり、質問の文脈は考慮していない。文脈に応じたふるまいは今後の課題であるといえる。

## 3.2 エージェントの協調

Venus & Mars での協調的なレシピ検索の例を以下に示す。

- (1) 個人エージェント: 「何でしょうか?」
- (2) ユーザ: 「豚肉を使った料理が食べたいです。」
- (3) レシピエージェント: (豚肉に反応して)「それでは豚肉を使った料理を調べて参ろう。こんなものありましたよ、どんなもんかな。」(検索結果をブラウザに表示する)
- (4) 健康エージェント: (豚肉に反応して)「豚肉は脂肪も少なめで消化もよい。風邪の時のビタミン補給にはうってつけじゃ。」
- (5) レシピエージェント: 「検索結果が多いようです。もっと絞り込みましょう。どのような味付けがお好みですか?」
- (6) 個人エージェント: (味付けに関する質問に反応して)「北村さんは前回『和風が好き』とおっしゃっていました。」
- (7) レシピエージェント: (和風に反応して)「それでは豚肉を使った和風料理を調べて参ろう。こんなものありましたよ、どんなもんかな。」(新しい検索結果をブラウザに表示する)
- (8) 個人エージェント: (ユーザのクリックに反応して)「何でしょうか?」
- (9) ユーザ: 「風邪に効く料理が食べたいです。」
- (10) レシピエージェント: 「?」
- (11) 健康エージェント: (風邪に反応して)「風邪には長ネギが良いんじゃないよ。」
- (12) レシピエージェント: (長ネギに反応して)「それでは長ネギを使った料理を調べて参ろう。こんなものありましたよ、どんなもんかな。」(長ネギを含んだ検索結果をブラウザに表示する)

Venus & Mars ではエージェントは 2 つの方法で協調する。1 つ目は上記の (5) ~ (7) のステップで絞り

†8 <http://www.goo.ne.jp>

†9 「材料」を含み「専門」と「商品」を含まないページ、あるいは「大さじ」を含むページを検索することを意味している。

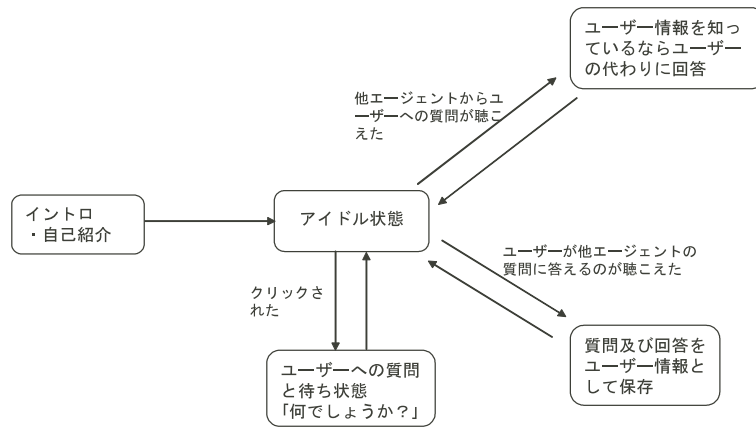


図 6 個人エージェントの状態遷移グラフ

込み検索を行うときに個人エージェントがレシピエージェントを手助けしているものである。レシピエージェントが絞り込み検索を行う味付けを尋ねると、個人エージェントがユーザのインタラクションの履歴を参照してユーザの代わりに「北村さんは前回『和風が好き』とおっしゃっていました。」と発話する。もちろん、ユーザが個人エージェントの提案が気に入らなければダイアログボックスを通してレシピエージェントに直接好みを伝えることができる。個人エージェントはこの会話を検知してローカルデータベースに記録していたユーザの好みを更新する。このように個人エージェントはユーザの会話を監視することでユーザの好みを学習する。

2 つ目は上記の (9) ~ (12) のステップで健康エージェントがレシピエージェントを手助けしているものである。レシピエージェントは健康についての知識を持たないため「風邪に効く料理が食べたい。」という発話に回答することができない。一方、健康エージェントは料理の食材と健康についての知識を持っているため「風邪には長ネギが良いんじゃないよ。」と発話する。レシピエージェントはこの発話を「長ネギ」というキーワードによる検索のきっかけとする。この例は、Venus & Mars ではエージェントの組み合わせを変更することで、様々な種類の協調的情報検索を実現できる能力を持っていることを示している。例えば名産品エージェントを加えれば、ユーザの「大阪の名物料理を食べたい。」という発話に対して、大阪

の名産品の食材を利用したレシピ検索を行うことができる。

#### 4 Wizard of Oz 法による評価

本章では Wizard of Oz 法を用いた MCI の評価について述べる。

##### 4.1 Wizard of Oz 法

MCI の評価に Venus & Mars をそのまま用いることは困難であった。それは現在の Venus & Mars はプロトタイプの段階であり、特に人間のユーザとスムーズに会話を行うことはできず、ユーザを長い間システムに注目させておくことが難しかった。そこで我々は、Wizard of Oz 法 [4][14][22] を用いることにした。

Wizard of Oz 法はウィザードと呼ばれる人間がシステムの一部または全体をシミュレーションすることで、見かけ上はコンピュータによるシステムとして動作させ、被験者の行動を観測する手法である。この実験では以下に示すように、ウィザードがキャラクタを通して被験者とインタラクションを行うように Venus & Mars を改造し、評価を行った。

- 情報検索における複数のキャラクタの効果を求めることが評価の目的であったので、個人エージェントは用いなかった。その代わりに名産品エージェントを用いた。名産品エージェントはローカルデータベースに日本の地名とその地域の



- 名産食材についての知識を持っている．地名や食材に関連するキーワードを含んだ発話に対して，「北海道の名産品はジャガイモがありますよ．」のように地名とその名産品を示すコメントを発話する．このように地名からレシピ検索へのきっかけを与えることでレシピエージェントと協調する．
- 名産品エージェント，レシピエージェント，健康エージェントのそれぞれに一人のウィザードを割り当てる．ウィザードはそれぞれの情報源を操作してを用いて被験者からの質問に日本語で応答する<sup>†10</sup>．ウィザードはその特定の領域の質問以外にも，エージェント自身についての質問（例えば「あなたは何歳ですか？」）や一般的な挨拶にも応答する．それ以外の質問に対しては「すみません．分かりません．」とだけ応答する．
  - エージェント（チャット）は起動された時点で自己紹介を行い，自分の能力を被験者に対して説明する．
  - ウィザードは被験者とは別室にあり，チャットシステムのようなインタフェースを通してのみ被験者からの質疑に答える．ウィザードは被験者の発話がどのウィザードに対してのものであるのかはキャラクターに付随する ID 番号によって認識できる．また被験者が他のウィザードがどのような会話をしているかも認識することができる．
  - 実験開始前に被験者にはウィザードの存在を知らせていないが，実験後にはそのことを知らせている．

†10 名産品エージェント担当のウィザードが用いているデータベースは「あのまちこのまち」(<http://www.gds.ne.jp/>)という Web サイトからその地名と名産品との関係を抜き出すことにより作成した．その数は約 3300 項目ある．レシピエージェント担当のウィザードは 3.1.1 節で述べた検索隠し味を用いた検索を手動で行った．健康エージェント担当のウィザードは「食育大事典」(<http://www.shokuiku.co.jp/>)という Web サイトを参照することにより健康と食材の関係に関するコメントを行った．

#### 4.2 インタフェースの比較実験

情報統合システムのインタフェースとして MCI は，(1) 複数のキャラクターが登場する，(2) キャラクターが互いに協調する，(3) キャラクターはそれぞれ特定領域の情報のみを扱うという役割分担がある，という三つの特徴を持っている．そこでこれらの特徴がユーザの情報検索におけるふるまいにどのような影響を与えるかを調査した．そこで表 2 に示すように 5 つの異なるインタフェースを用いた．

- (A) 協調型マルチ： Venus & Mars に最も近くなるように実装したシステムである．名産品エージェント，レシピエージェント，健康エージェントの 3 体のキャラクターが登場する．ウィザードは Venus & Mars のように，それぞれのキャラクターがお互いに協調しながらレシピページを検索するようにキャラクターを操作する．
- (B) シングル万能： ブラウザ上には万能キャラクター 1 体だけ登場する．ウィザードはそれぞれ自分の担当エージェントの知識の領域に関する発話に対して，この 1 体のキャラクターを通して応答する．また，ウィザードの一人は，挨拶やキャラクター自身への質問にも対応する．このようにすることで，このキャラクターは三つのエージェントの全ての能力を持ちながら，被験者は単体のエージェントと話しているように感じる．
- (C) チャット形式： キャラクターが登場せず，チャット形式でレシピ検索を行う．それ以外はシングル万能と同様である．
- (D) 非協調型マルチ： キャラクター同士の協調がないものである．3 体のキャラクターが登場するが，話しかけられたキャラクターを担当するウィザード以外は応答しないため，キャラクター同士の協調は行われない．例えば名産品エージェントに対して健康に関する発話がされた場合には，そのウィザードは「すみません，分かりません．」と応答し，健康エージェントのウィザードは何も応答しない．
- (E) 協調型マルチ万能： キャラクター間の役割分担がないものである．万能キャラクターが 3 体登場し，そのそれぞれが名産品，レシピ，健康のどの

表 2 実験に用いるインタフェース

	キャラクタ数	協調	役割
A 協調型マルチ	3	有	有
B シングル万能	1	-	-
C チャット形式	0	-	-
D 非協調型マルチ	3	無	有
E 協調型マルチ万能	3	有	無

発話に対しても応答する。さらに他のキャラクタの発話に対しても協調的に動作する。協調のさいにウィザード同士が混乱しないように、発話したキャラクタに対してどのキャラクタが協調的な発話をするのかをあらかじめ設定している。

これらの5種類のインタフェースに対して、それぞれ5人ずつ、合計25人の大学生を被験者として割り当てた。それぞれの被験者の最初の20回の発話に対し、その発話を名産品、レシピ、健康、キャラクタ、その他に分類した。

#### 4.3 実験結果

キャラクタの数の違いによる比較を行うために、協調型マルチ、シングル万能、チャット形式における被験者の挙動を比較した。その結果を表3に示す。

その結果、有意水準を10%以下として優位な差として得られたものは以下の通りである。

- シングル万能はチャットよりもその他に関する話題が少ない（有意水準3%）
- シングル万能はチャットよりもキャラクタに関する話題が多い（有意水準7%）
- 協調型マルチはシングル万能よりもレシピに関する話題が少ない（有意水準2%）
- 協調型マルチはシングル万能よりも健康に関する話題が多い（有意水準6%）
- 協調型マルチはチャットよりも名産品、健康に関する話題が多く、レシピに関する話題が少ない。（有意水準1%以下）
- 協調型マルチはチャットよりもキャラクタに関する話題が多い（有意水準10%）

以上の結果からキャラクタが存在しないチャット形式では被験者の検索がレシピ関連に集中しており、当

然のことながらキャラクタ自身に対する質問は一切行われていないことがわかる。またキャラクタ数が増加するにしたがって、会話の内容がレシピに関するもののみから名産品や健康へと話題が広がっていることが分かる。

次に、キャラクタ間の協調の有無による比較を、協調型マルチと非協調型マルチで、またキャラクタ間の役割分担の有無による比較を、協調型マルチと協調型マルチ万能をそれぞれ比較することにより行った。その結果を表4に示す。

しかしながら、この評価に関してはいずれも特に大きな有意差は認められず、キャラクタ間の協調の有無と役割分担の有無によるユーザのふるまいへの影響は今回の実験からは立証できなかった。したがって今回の評価ではキャラクタ間の協調や役割分担よりも、キャラクタの数そのものがユーザのふるまいに変化を与えるものであることが明らかになった。

これはすなわち目に見えるキャラクタの存在そのものが利用者との会話を引き起こし、それが話題の多様性へとつながってゆくものと思われる。キャラクタ間の協調や役割分担の有無が重要な変化をもたらさなかったことに関しては、本評価実験では被験者が自由に情報検索する設定にしていたので、その違いが際立つほど会話自体が複雑にならなかったためであると考えられる。今後は被験者に明確な目標を伴う情報検索タスクを与え、そのタスク達成に対する被験者の挙動を観測することが必要であると考えられる。これは今後の課題としたい。

#### 5 関連研究

検索エンジンや情報エージェントを組み合わせたり、統合したりすることはそれぞれのシステムにさらなる

表 3 キャラクタの数の違いによる平均発話回数の比較

チャット/シングル万能				
分類	チャット	シングル万能	t-値 (n=8)	有意水準
名産品	0.2	1.4	-1.26	0.12
レシピ	14.8	13.4	0.88	0.20
健康	0.4	1	-0.8	0.22
キャラクタ	0	1.8	-1.61	0.07
その他	4.6	2.4	2.17	0.03
シングル万能/協調マルチ				
分類	シングル万能	協調マルチ	t-値 (n=8)	有意水準
名産品	1.4	2.8	-1.18	0.13
レシピ	13.4	10	2.42	0.02
健康	1	2.4	-1.72	0.06
キャラクタ	1.8	1.6	0.12	0.45
その他	2.4	3.2	-0.70	0.25
チャット/協調マルチ				
分類	チャット	協調マルチ	t-値 (n=8)	有意水準
名産品	0.2	2.8	-3.41	0.004
レシピ	14.8	10	3.63	0.003
健康	0.4	2.4	-3.08	0.007
キャラクタ	0	1.6	-1.37	0.10
その他	4.6	3.2	1.10	0.15

表 4 キャラクタの協調と役割りの違いによる平均発話回数の比較

非協調マルチ/協調マルチ				
分類	非協調マルチ	協調マルチ	t-値 (n=8)	有意水準
名産品	2	2.8	-0.78	0.22
レシピ	12.4	10	1.39	0.10
健康	2.6	2.4	0.14	0.44
キャラクタ	0.2	1.6	-1.18	0.13
その他	2.8	3.2	-0.35	0.36
協調マルチ万能/協調マルチ				
分類	協調マルチ万能	協調マルチ	t-値 (n=8)	有意水準
名産品	1.4	2.8	-1.56	0.07
レシピ	9.8	10	-0.12	0.45
健康	1.4	2.4	-1.10	0.15
キャラクタ	2.2	1.6	0.30	0.38
その他	5.2	3.2	1.54	0.08

価値を加える。MetaCrawler[17]や SavvySearch[8]のようなメタ検索エンジンは複数の検索エンジンの結果を統合することでその性能を改善することに成功している。BIG[13]は、情報の質とその収集に要する時間やコストとのトレードオフを考慮することで、複数の情報源から効率的に情報収集する情報エージェントである。RETSINA[18]はインタフェースエージェント、タスクエージェント、リソースエージェントの3種類の情報エージェントから構成されるマルチエージェントシステムである。インタフェースエージェントはユーザからの要求を受け取り、得られた結果を提示するためにユーザとインタラクションを行う。タスクエージェントは他のエージェントと情報交換を通して特定領域のタスクを解決する。リソースエージェントは異なる種類の情報源へのアクセスを提供する。また、連邦システム[6]、InfoSleuth[2]、LARKS[12]なども情報ブローカリングや情報マッチメイキングメカニズムを組み込んだ情報配送のためのマルチエージェントシステムである。

上記のような従来の協調型情報統合システムでは、情報エージェントや情報源の組み合わせはシステム的设计者によって決定され、ユーザから隠されていた。そのためユーザは固定的なインタフェースに要求を出し、その結果を受けとることしかできず、情報エージェントの組み合わせや協調メカニズムを変えることはできなかった。

例えば RETSINA エージェントは再利用可能であり、そのインタフェースはシステム設計者にオープンであるがユーザに対してはオープンであるとはいえない。ユーザはインタフェースエージェントを通してのみシステムにアクセスできるだけである。連邦システムではエージェントの協調は KQML[5]のような ACL(Agent Communication Language)で記述される。ACLは情報エージェントにオープンなインタフェースを提供するが、人間のユーザが ACLを使ってエージェントと直接コミュニケーションをするのは困難であり、人間のユーザに対してはオープンとは言えない。このように、従来のシステムはユーザに対してオープンなインタフェースを提供するようには設計されていなかった。

それぞれのユーザは情報検索や情報統合に対して異なった要求や好みを持っている。あるユーザはある特定の検索エンジンを好み、別のユーザはまた別の検索エンジンを好むかもしれない。このように既存の固定的な情報統合システムよりも、ここではむしろ、気に入った情報エージェントの組み合わせを柔軟にカスタマイズでき、ユーザと協調的に作業できるようなフレームワークの実現を目標としている。そのための第一ステップとして、キャラクタインタフェースをもつ複数のエージェントが他のエージェントやユーザと協調的に動作する MCIを開発した。

複数のキャラクタを用いた情報システムの研究としては以下のものがあげられる。Andre と Rist の Inhabited Market Place[1]は主にプレゼンテーションメディアにおけるマルチキャラクタの利点を強調している。すなわちそれぞれのキャラクタが独自の視点から商品に対するコメントを発することによりプレゼンテーションの効果を強めている。Prendinger と Ishizuka による Let's Talk[16]は英語教育システムにキャラクタエージェントを応用したものである。ここではキャラクタエージェントはエージェント同士の社会的関係に基づく感情表現の問題を扱っている。本論文ではキャラクタエージェントの感情表現は扱っていないが、ユーザとのより高度で親密なインタラクションのためには感情表現も重要になるであろう。また高橋と武田による TelMeA[19]はコミュニケーションメディアとしての複数キャラクタエージェントの効果を示している。ここでは従来のチャットや BBS などのテキストベースのコミュニケーションメディアにキャラクタを付加することで、発言者の識別に対して効果を持つことを示している。

またアーキテクチャの視点から比較すると、上記のシステムでは一つのマシンで複数のエージェントを制御する集中型のアプローチを採用しているのに対して、MCIでは、エージェントの本体部は個々のサーバに分散的に存在し、エージェントの頭部がクライアントマシン上でインタラクションを行う分散型アプローチを採用している。このように MCI はエージェントがインターネット上で物理的に分散したマルチエージェントシステムであるといえる。

## 6 まとめ

本論文では、キャラクタインタフェースを持つ複数のエージェントの協調による WWW 情報統合システムを議論した。その実現のために、キャラクタが他のキャラクタやユーザと協調するマルチキャラクタインタフェース (MCI) を提案し、その応用プロトタイプとして Venus & Mars という協調型レシピ検索システムを開発した。MCI はキャラクタという親しみやすいインタフェースを持ち、複数のキャラクタのインタラクションを通してユーザは情報統合の過程を理解することができる。さらに、エージェントの組み合わせや挙動をユーザ自身が柔軟に変更できるという特徴を持つ。また MCI の評価を Wizard of Oz 法を用いて行い、キャラクタの数、キャラクタ同士の協調の有無、キャラクタの役割の有無に応じてユーザのふるまいがどのように変化するかを調査した。その結果、複数のキャラクタが登場することにより、ユーザが広い範囲の情報検索に興味を持つようになることが明らかになった。

現在のプロトタイプで実現した協調タスクは単純なものであり、より高度な協調エージェントを作るためには以下に示すように、解決すべき多くの課題がある。

**協調能力：** エージェントはいったんクライアントマシンに呼び出されるとキャラクタを介してプラグアンドプレイでお互いに協調する。現在のプロトタイプにおけるエージェントの協調動作は、検索キーワードの連想や補完といった単純なものである。より高度な協調的特徴を実現するためにはマルチエージェントシステムの分野で研究されている協調技術や交渉技術を使った協調機構 [20] を組み込む必要がある。

**表現能力：** 現在のプロトタイプではエージェント間の協調は会話によって表現される。コンピュータのディスプレイ上で行われるエージェント間の会話は一時的なものであり、複雑な協調動作を表すには適していないかもしれない。このことから、例えばエージェントが他のエージェントだけでなく仮想の対象物に対してもインタラクション

できるような仮想空間を作ることによって、表現能力を改善する必要がある。そのような能力は Andre らによって行われている研究 [1] と関連が深い。

**会話能力：** 現在のプロトタイプではエージェントは主に自然言語によってユーザとインタラクションするが、その認識はユーザの発話からキーワードを抽出するといった単純なものである。自然言語は、人間のユーザ、特に子供やお年寄りのような初心者ユーザにとって最も自然なコミュニケーション方法である。ユーザとのより複雑なインタラクションを実現するためには、高度な自然言語処理技術がエージェントに導入される必要がある。

現在のインターネットベースの情報システムは Web 技術に大きく依存している。Web ページにはテキストだけでなく画像や音声も貼り付けることができるため、非常に表現力豊かである。そして XML や Semantic Web 技術は現在の Web 技術をさらに高度なものにすると考えられる。その一方で、現在の Web 情報システムはページ単位で情報を提供する受動的なシステムである。キャラクタは XML 技術とはまた異なる方法で Web 技術を進化させ、より能動的にインタラクティブに見えるようにする。その意味において、MCI はキャラクタを介して複数の情報源を協調的に統合するフレームワークを提供している。これは XML 技術と共存させることにより新しいエージェントベース情報統合システムへと発展してゆくことが期待される。

## 謝辞

本研究はイメージ情報科学研究所および文部科学省特定領域研究「情報洪水時代におけるアクティブマインニングの実現」からの援助を受けて行われました。本研究に関して貴重なコメントをいただきました京都大学石田亨教授ならびに匿名の査読者に対して感謝の意を表します。

## 参考文献

- [1] Andre, E. and Rist, T.: Adding Life-Like Synthetic Characters to the Web, in *Cooperative Infor-*

- ation Agents IV, Klusch, M. and Kerschberg, L. (eds.) Lecture Note in Artificial Intelligence 1860, Springer, 2000, pp. 1-13.
- [ 2 ] Bayardo Jr., R. J., Bohrer, B., Brice, R. S., Cichocki, A., Fowler, J., Helal, A., Kashyap, V., Ksiezyk, T., Martin, G., Nodine, M. H., Rashid, M., Rusinkiewicz, M., Shea, R., Unnikrishnan, C., Unruh, A. and Woelk, D.: InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments, in *Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD 1997)*, ACM Press, 1997, pp. 195-206.
- [ 3 ] Cassell, J., Sullivan, J., Prevost, S. and Churchill E. (eds.): *Embodied Conversational Agents*, MIT Press, 2000.
- [ 4 ] Dahlback, N., Jonsson, A. and Ahrenberg, L.: Wizard of Oz studies - Why and How, in *Proceedings of the International Workshop on Intelligent User Interfaces*, 1993, pp. 193-200.
- [ 5 ] Finin, T., Labrou, Y. and Mayfield, J.: KQML as an Agent Communication Language, in *Software Agent*, Bradshaw, J.M. (ed.), AAAI Press, 1997, pp. 291-316.
- [ 6 ] Genesereth M.R.: An Agent-Based Framework for Interoperability. in *Software Agent*, Bradshaw, J.M. (ed.), AAAI Press, 1997, pp. 317-345.
- [ 7 ] Hearst, M.A.: Information Integration, *IEEE Intell. Syst.*, Vol. 13, No. 5 (1988), pp. 12-24.
- [ 8 ] Howe, A.E. and Dreilinger, D.: Savvy Search: A Metasearch Engine That Learns Which Search Engines to Query, *AI Mag.*, Vol. 18, No. 2 (1997), pp. 19-25.
- [ 9 ] Ishida, T.: Interaction Design Language Q: The Initial Proposal, in *Proceedings 15th Annual Conference of Japanese Society for Artificial Intelligence*, 2A2-03, 2001.
- [10] 石川幹人: サイコロと Excel で体感する統計解析, 共立出版, 1997.
- [11] Klusch, M. (ed.): *Intelligent Information Agents*, Springer, 1999.
- [12] Klusch, M.: Information Agent Technology for the Internet: A Survey, *J. Data Knowl. Eng.*, Vol. 36, No. 3 (2001), pp. 337-372.
- [13] Lesser, V. R., Horling, B., Klassner, F., Raja, A., Wagner, T. and Zhang S. XQ.: BIG: An Agent for Resource-bounded Information Gathering and Decision Making, *Artif. Intell.*, Vol. 118 (2000), pp. 197-244.
- [14] Maulsby, D., Greenberg, S. and Mander, R.: Prototyping an Intelligent Agent through Wizard of Oz, in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1993, pp. 277-284.
- [15] Oyama, S., Kokubo, T., Ishida, T., Yamada, T. and Kitamura, Y.: Keyword Spices: A New Method for Building Domain-Specific Web Search Engines, in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, 2001, pp. 1457-1463.
- [16] Prendinger, H. and Ishizuka, M.: Let's Talk! Socially Intelligent Agents for Language Conversation Training, *IEEE Trans. Syst., Man Cybern., Part A*, Vol. 31, No. 5 (2001), pp. 465-471.
- [17] Selberg, E. and Etzioni, O.: Multi-Service Search and Comparison Using the MetaCrawler, in *Proceeding of the 4th International World Wide Web Conference*, 1995, pp. 195-208.
- [18] Sycara, K. and Zeng, D.: Coordination of Multiple Intelligent Software Agents. *Int. J. Coop. Inf. Syst.*, Vol. 5, No. 2&3 (1996), pp. 181-211.
- [19] 高橋徹, 武田英明: TelMeA: 非同期コミュニケーションシステムにおける avatar-like エージェントの効果と web ベースシステムへの実装, 電子情報通信学会論文誌, Vol. J84-D-I, No. 8 (2001), pp. 1244-1255.
- [20] Weiss, G. (ed.): *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.
- [21] Wiederhold, G.: *Intelligent Integration of Information*, Kluwer Academic, 1996.
- [22] Yang, Y., Okamoto, M. and Ishida, T.: Applying Wizard of Oz Method to Learning Interface Agent, in *Proceedings of Workshop on Software Agent and its Applications (SAA2000)*, 2000, pp. 223-230.