

# 自立的な対話をする キャラクタエージェントシステム

守田 航大 角 薫  
(公立はこだて未来大学大学院)<sup>†</sup>

## 1 はじめに

これまで我々は「サービスマインドを伝えるエージェント・トレーニングシステムの研究」<sup>[1][2][3]</sup>の研究を行ってきた。この研究は、エージェントと対話していくことによって接客業に対するトレーニングを受けることができるシステムである。この研究では主に外装である Kinect 等を用いたインターフェースについての研究を行っていた。本研究では内装であるエージェントの台詞についての研究を行っている。現在、エージェントの台詞は予め開発者が設定した言葉しか喋らず、決まった入力に対して決まった返ししかできない。しかし、本研究では、はじめは必要最低限の知識や情報のみを所有し、その情報を元にユーザとの対話を繰り返し、エージェントが対話の中で成長していくようなシステムの開発を目指している。方法としてはインターネットを辞書のように扱い、知識を蓄えていき、その知識を元にエージェントの台詞を構築する。これにより、予め台詞を設定する必要はなく、さらに様々な場面に対応することができる。システムの開発後は、実験を行いエージェントの台詞に対しての評価をする予定である。

## 2 これまでのキャラクタエージェント対話システム

これまでの研究では、接客の心得であるサービスマインドに着目し、エージェントと対話していくことにより接客の教育トレーニングを受けられるシステムを開発してきた。日本ではあまり親しみがないが、海外ではエージェントの研究は進んでおり<sup>1</sup>、すでに実践で投入されているものもある<sup>2</sup>。本研究では、海外でも評判の高い日本の接客サービスの無形の文化をシステムに取り入れることで場所や時間を問わず学習が可能となる。このシステムでは、音声認識システム、音声合成システム、対話システム、3D モデルのキャラクタエージェント表示

システムからなるキャラクタ対話システムを利用していた<sup>[4][5][6][7][8]</sup>。

対話システムでは AIML を Artificial Intelligence Markup Language(以下 AIML)を使用した。AIML は自然言語ソフトウェア構築のための XML を応用したマークアップ言語であり、AIML はテンプレートが自由に変更できるため使用された。また AIML ではトピックという機能があり、話の内容ごとに会話を分けることができる。例えば、同じパターンであっても、異なるトピックであれば違ったテンプレートを返し、互いに干渉されることもない。これまでの我々の研究では接客業について教えるテーマ事に分けて作成している。3D モデルのキャラクタエージェントは笑い・悲しみ等の表情と母音の口の形を用意しモーフィングを使用することによりなめらかな表情と口の動きを提示した。さらに Kinect を用いた表情認識及びジェスチャ認識を開発し、実装した。

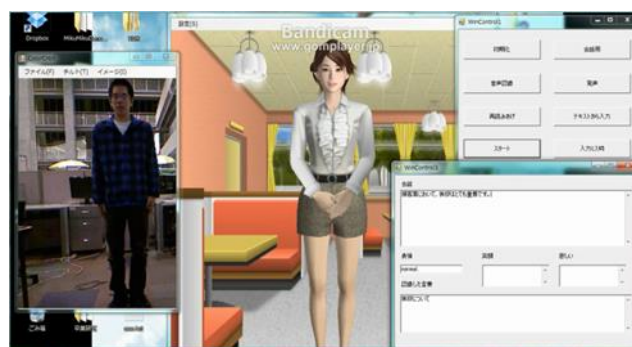


図1 サービスマインドを伝えるエージェント  
トレーニングのスクリーンショット

表情認識では Kinect for Windows SDK の Face Tracking という機能、Face Tracking Basics-WPF サンプルをベースに、インターフェースに合うように組み込んである。サンプルでは顔を検出するのみであるが、Kinect for Windows の Face Tracking には Animation Units/AU という概念があり、これが表情を特徴づけるデータになっている。具体的には、主に眉の角度、口角の角度などを基準として判別しており、この Animation Units の特徴量をそのまま使用している。口を横に引き伸ばしている状態(AU2 が 0.4 以上)で笑顔と判定し、それ以外で眉をし

<sup>1</sup> Sense.ly: DocPal,  
<http://www.sense.ly/index.php/applications/>

<sup>2</sup> SimCoach, <http://www.simcoach.org/>

\*[g2114031@fun.ac.jp](mailto:g2114031@fun.ac.jp), [kaorus@fun.ac.jp](mailto:kaorus@fun.ac.jp)

<sup>†</sup> 函館市亀田中野町116番地2 公立はこだて未来大学大学院システム情報科学研究科

かめている状態(AU3 が 0.4 以上)で悲しい顔と判定している。また、口が開いている(AU1 が 0.5 以上)の場合は除外している。口が開いていると笑顔とは認識されにくく、本システムでは音声認識と同時に進んでいるため口が開いてしまう。よって AU1 が 0.5 以上は除外している。

ジェスチャ認識では、Kinect for Windows SDK の骨格検出をベースにしている。取得している骨格のポイントを、骨格と同時に取得している RGB 映像での座標に照らし合わせ、どの程度骨格の座標に変化があるかを読み取ってジェスチャ認識を行っている。Kinect の骨格認識では頭、肩、右手、左手、腰、右足、左足など計 20 箇所の骨格ポイントを検出することができるが、本研究のジェスチャ認識では頭、背骨、腰の三か所の骨格のポイントを使用して認識している。認識できるジェスチャは「浅いお辞儀」と「深いお辞儀」を検出できるようにしており、「浅いお辞儀」では約 15 度の角度でお辞儀と検出され、「深いお辞儀」では約 30 度以上の角度で深いお辞儀と検出されるようにした。

接客トレーニングシステムでは、褒め方についてとクレームについてと挨拶についての 3 つのパートに分かれている。褒め方についてのトレーニングでは表情認識に加え、浅いお辞儀のジェスチャ認識を行っている。クレームについてのトレーニングでは表情認識と深いお辞儀のジェスチャ認識を行っている。挨拶については笑顔の表情トレーニング及び浅いお辞儀のトレーニングを行っている。トレーニングの流れとして図 2 のような形の対話がある。進む流れを矢印で表しており、青い矢

印は音声認識・表情認識・ジェスチャ認識全てが成功した場合(トレーニングを終了した場合)に進む先を表しており、赤い矢印は声認識・表情認識・ジェスチャ認識のいずれかが失敗した場合の進む先を表している。初めはエージェントの自己紹介と、接客業について何を聞きたいかとユーザに促すパートである。その下にそれぞれ「お客の褒め方」と「クレームの対応」と「基本的な挨拶」についての接客トレーニングのパートを作成してある。この褒め方とクレームと挨拶はエージェントがどのトレーニングを行うかを問い、ユーザがいずれかを答えることによって会話パートに進むことになる。会話パートでは接客に関する本<sup>[9]</sup>を参考にして内容を作成されている。褒め方には褒めるテクニックと笑顔のトレーニングを、クレームについては対処方法と悲しい顔と深いお辞儀のジェスチャを、挨拶ではどのお店でも使う基本的な挨拶についてと笑顔と浅いお辞儀のトレーニングをすることができる。トレーニングではまずエージェントがお手本となる表情を見せ、次にユーザがトレーニングするという流れである。トレーニングでは、エージェントのいるウィンドウ下方にユーザが話す台詞が書いてあるオレンジ色のバーが現れる。その際、ユーザは表情を作り、ジェスチャをしながらその文を読むといったものである。この際、読み上げる文に関しては音声認識と AIML を使用して喋れているのか判断をし、表情とジェスチャは Kinect によって判断をする。

被験者を用いた評価実験によりシステムの効果を評価した。評価の結果、接客のサービストレーニングシス

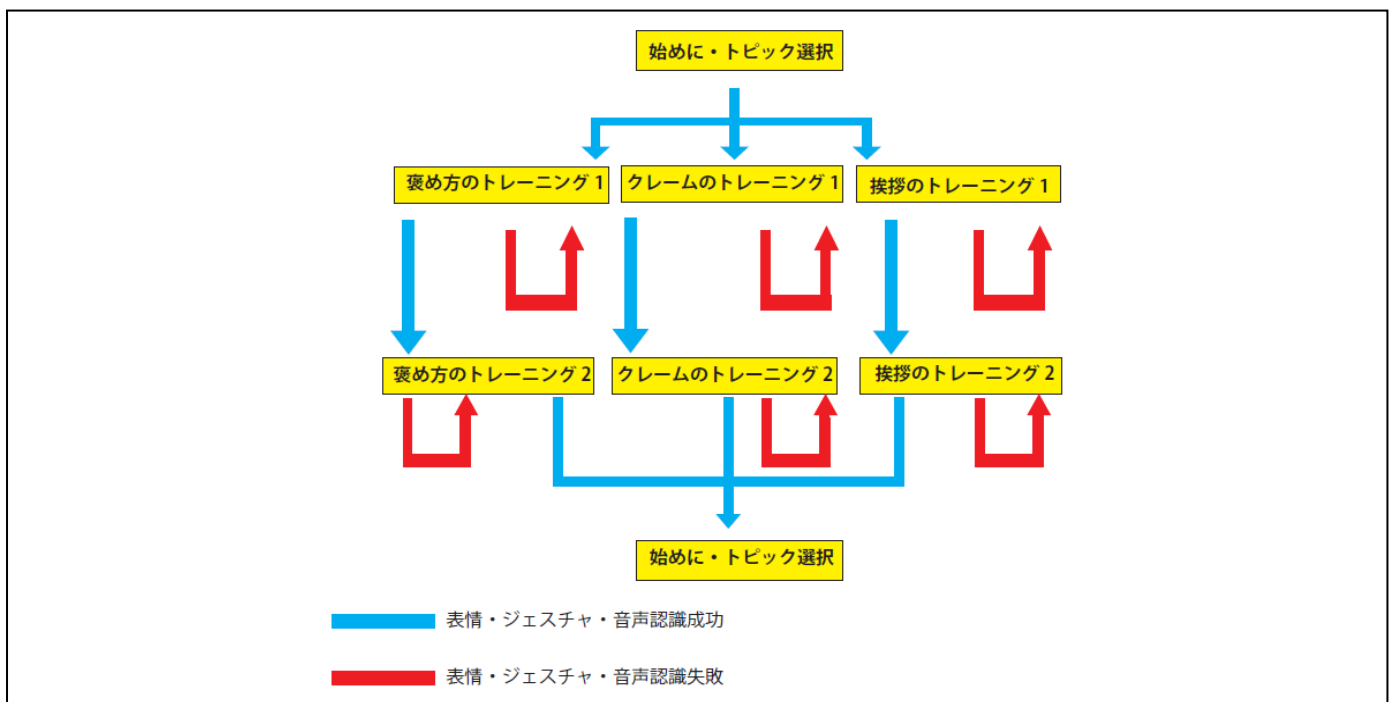


図 2 トレーニングの流れ

テムとして過半数のユーザに評価が高かったことがわかった

これまで、このようなエージェントを用いた対話システムを開発してきたが、前もって様々なパターンの受け答えを用意しなくてはならなかった。本研究では、前もって受け答えを用意しなくても Web 上のデータを利用することにより自動的に受け答えを学習する仕組みを考えた。

### 3 関連研究

iOS スマートフォン向けのユーザ支援アプリケーションに Siri がある<sup>3</sup>。音声認識を用いて、音楽をかけたり天気予報を調べたり、Web ブラウザで検索をかけたりと手で操作しなくても音声認識で様々な機能を引き出してくれる。スマートフォンの機能を引き出すだけでなく、適当な言葉で話しかけると、一般的な会話として成立する応答を返している例も見かけられる。本研究とは、ブラウザの検索機能が似ている。

Web 上のデータをエージェント同士の対話にする研究がある<sup>[10]</sup>。この研究は、ウェブのコンテンツを二人のエージェントの対話でより高度な情報としてユーザに提示する。Web コンテンツの主題語を抽出し、その主題語についての会話をしていく。片方のエージェントがコンテンツの主題的な内容を話し、もう一人のエージェントがその説明に対して補足的な質問をする。例として、説明しているエージェントの台詞に出てきた人物や物事について「〇〇って誰?」「〇〇って何?」等の質問をする。説明エージェントはそれらの質問に対して Web

やそのページの中からの解答を得て答えを返す。この質問があることでユーザにより高度な情報を提供できる。本研究とは、Web の情報を読み込む技術が類似している。

世界で初めてチューリングテストに合格したスーパーコンピュータ Eugene がある<sup>4</sup>。Eugene は Turing Test 2014 に合格した。チューリングテストとは、評価者の相手がわからない状態でテキストチャットを行い、評価者が「相手は人間か、コンピュータか」を判別するテストである。もし、判定者の 30%以上から「この相手は人間かコンピュータか判別できない」と評価されれば合格だとされており、Eugene は判定者の 33%から「人間かコンピュータかを判別出来ない」と評価された。Eugene は自然な対話をしており、そのテクニックの 1 つとして会話の主導権を握るような手法を用いている。本研究ではこの会話の主導権を握る手法を参考にしている。

### 4 自立的な対話をするキャラクタエージェントシステム

開発に使用した言語は 3D モデルのアニメーションを用いるために C# (Visual Studio 2010) と XNA を使用する。この XNA は 3D モデルを使用するのに長けており容易に 3D モデルを表示、利用できるものである。XNA では Xbox360 のゲームの開発をする事もできる。

これまでの我々の研究ではエージェントは会話パターンの元、ユーザとの対話を果たしていた。会話パターンとは「ある決まった言葉(入力)に対して、決まった応答(出力)をする」ものである。しかし、本研究では、会

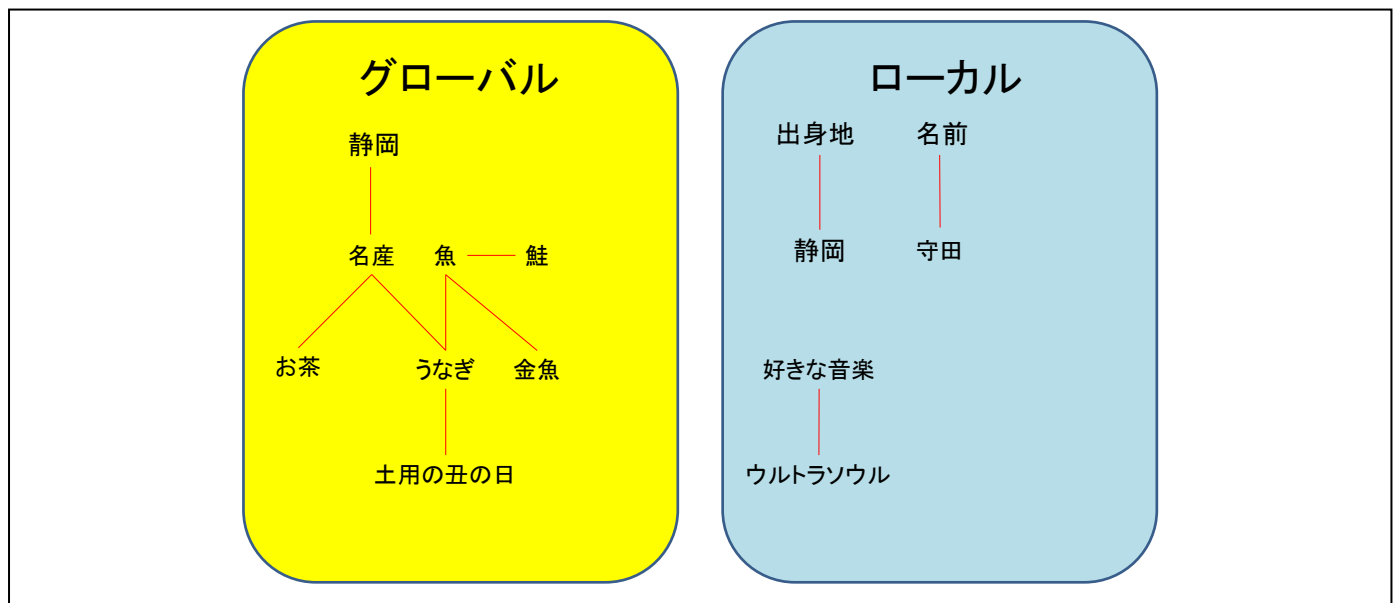


図3 情報の保持

<sup>3</sup> Siri: <https://www.apple.com/jp/ios/siri/>

<sup>4</sup> Eugene: <http://www.princetonai.com/bot/>

話パターンでの対話ではなく、会話の中で情報を得て成長し対話するエージェントの作成を目指す。初めは必要最低限の情報のみを保持し、その情報を元にユーザと会話をしていく。エージェントは会話の中で必要な情報を得て成長していく。つまり、対話すればするほどエージェントの所持する情報が増えていく。

情報保持の方法として、本研究ではグローバルな情報とローカルな情報を区別する手法をとる。会話の中にはグローバルな話とローカルな話がある。グローバルな話は一般的な話で、天気や小説等の社会の中で認知され共有されている情報を元にした会話である。対してローカルな話では、話者(ユーザ・エージェント)の身の回りの個人的な情報を元にした会話であり、その話者の周囲の者にしか認知されていない。例として、「かえで」という名前の人物がいたとして、その人物の中では「楓」という<植物の名前>というグローバルな情報と「かえで」という<自分の名前>というローカルな情報を持ち合わせている。グローバルな情報のほとんどはインターネットを利用することで入手することができる。しかし、ローカルな情報は話者との会話の中でしか入手することができない。本研究では、自然な対話を目指したシステムの開発を目的としている。そのためにこのグローバルな情報とローカルな情報を分割して保持する必要があると考えた。グローバルな情報とローカルな情報を区別して記憶することで、ユーザの区別が簡単に行うことができ、複数のユーザに利用してもらうことによりグローバルな情報が得やすくなる。グローバルな情報を複数のユーザで共有することで、より多くの人から情報を得ることができるのでエージェントの知識が増していく。

自然な対話とするために、エージェントは先手で質問をする。先手で質問することで、エージェントは会話の主導権を握ることができ、自然な対話へとしやすくなる。チューリングテストに合格した Eugene の会話でも先手をとって質問をしていた。

質問する内容は以下の様なものを想定している。

- ・ 名前
- ・ 出身地
- ・ 家族構成
- ・ 苦手な食べ物
- ・ 好きな食べ物
- ・ 好きな音楽
- ・ 好きな映画
- ・ 好きな俳優

質問内容はユーザのローカルな情報でプロフィールを埋めるような形で質問をしていく。これらのことを質問することで会話のトリガーとし、基本情報として保持していく。さらにこの質問から得られた情報から会話を広げていく。

エージェント会話の中で、ユーザの応答と Web 検索を用いて情報を得ていく。ユーザとの会話の中で保持していない単語が出てきた場合、Web 検索を行い、情報を得ていく。例えば、エージェントが出身地を聞いた場合、ユーザの答えとして出身地がローカル情報として保持される。もし、その出身地の単語がグローバル情報に保持されていなかった場合、グローバル情報にも保持される。会話のトリガーとして最初に保持している質問群の中で特定の質問の応答に対して、グローバルな情報とローカル情報で分けて考える。例として、「好きな音楽・映画・小説」などはグローバルな情報とローカルな情報の2つに保持されるが、「家族構成」などはローカルな情報のみとして時保持されるように分類わけが出来る。

## 5 適用

Web 検索では出身地の事柄や、人物の事柄・作品の事柄等様々なことを検索していく。出身地であれば、「出身地名 名産」などで検索を行う。そうすることで、その名産品を新たな情報として保持し、エージェントの情報が広がると考えられる。作品の事柄、例えば小説のタイトルならば Web 検索を行い、その小説の作者や関連図書、その作者から作者が書いた別の小説を引用する。エージェントはこれらの引用した情報をユーザに対して提示することで、より知的なエージェントになると考えられる。

「出身地名 名産」等の検索は「Google Web API」を用いる。「Google Web API」を用いることで複数の Web ページを参照することができる。「Google Web API」は検索スピードが早く、有益な情報が上位より得られやすいと判断したので採用した。エージェントは「Google Web API」を使用した場合、検索結果から得られた上位数ページを参照して目的の情報を各 Web ページから取り出す。

小説のタイトルや音楽等の情報は「Wikipedia API」を用いて Web 検索を行う。「Wikipedia API」は Wikipedia によって無償で提供されており、膨大な情報が収められていることから使用することにした。Wikipedia は誰にで

も編集が可能という側面もあり情報としては信憑性に欠ける部分があると考えられるが、内容や人物に関する主観的な書き方に依る部分が多い。しかし、小説のタイトルやその作者や登場人物、音楽のタイトルやそのアーティスト・作曲者、映画のタイトルや監督、などの事実に関する部分が偽られて記載されている部分はほとんどない。Wikipedia ではこのような事実を情報として得るのには最も適していると考えられる。さらに、Wikipedia は誰にでも編集ができるので、情報の更新が早いことが特徴である。エージェントは Wikipedia において、関連情報を取得する。例えば、小説のタイトルを検索し、適切なページが見つかった場合、その小説の作者、その作者が執筆した別の作品を情報として得る。そしてユーザに提示する。音楽であった場合、音楽のタイトルからその楽曲のアーティスト・作曲者を情報として保持し、さらにそのアーティスト・作曲者が演奏、歌っている別の楽曲を提示する。

初めはユーザのプロフィールを埋めるような形でエージェントは情報を会得していく。その後、プロフィール欄が埋まってきた場合、保持している情報からユーザの興味があると推測できる情報を提示していくプロフィールではユーザの「好きなもの」を中心として質問していく。ユーザの好きなものであれば、その事柄に関連したものは興味があると推測できる。エージェントはこの興味がありそうな事柄について最新の情報を提示していく。最新の情報については Google のニュース検索を用いる。例えば、出身地が東京であった場合「東京」で Google ニュース検索をする。検索した結果、日付順で最も新しいニュースを幾つかユーザに提示することで最新的话题を提示する。

エージェントは初め、ユーザのプロフィールを埋めるような形で質問をしていく。このプロフィールを埋める質問はいくつか用意されており、エージェントはランダムに情報の格納されていないものを質問していく。会話の例として、はじめにエージェントがユーザの「出身地」を聞いた場合を想定する。

**エージェント**「あなたの出身地はどこですか？」

**ユーザ**「静岡県です」

この場合、エージェントはユーザの出身地に「静岡県」というローカルな情報を格納し、同時にグローバルな情報において静岡県が入っていないければ「地名」と「静岡県」を組み合わせる。情報が入っていないければ、

**エージェント**「静岡って何がありますか？」

と質問する。その後、返された言葉の中の単語を知識としてグローバルな情報に格納する。

**ユーザ**「うなぎが名産だね」

と答えたとする。ここでエージェントは「名産」や「有名」と言った言葉に反応し、その前の言葉を知識として追加する。今回の場合は「うなぎ」を追加する。ここでうなぎに対して知識が入っていなければ、

**エージェント**「うなぎってなんですか？」

と質問する。知識がすでに入っていた場合、「うなぎ」と「静岡県」をつなぎ合わせる。今回は知識が入っていなかった場合とする。質問に対してエージェントはユーザの答えから「うなぎ」とは何なのかを教えてもらいます。もし「魚」など単語であればそれを「うなぎ」とつなぎ合わせ、「うなぎは魚だよ」や「魚だよ」などの場合は最後の名詞を情報として保持する。

**ユーザ**「うなぎは魚だよ」

とここで答えたとする。ここでエージェントは「うなぎ」と「魚」をつなぎ合わせる。そして、

**エージェント**「そうですか。静岡県の名産は他にはお茶があるようです」

といった言葉を返す。この時、「出身地（今回は静岡県）名産」と Google Web API を用いてインターネットで検索する。検索結果から得られた候補から情報の記されているページにアクセスし、該当の情報を得てくる。「静岡県 名産」と検索した結果から得られたページからお茶という解答が得られた場合、エージェントはこのような答えを返す。その後、エージェントは今回得られた情報からランダムに言葉を選定し、選ばれたものの最新のニュースをユーザに対して提供する。今回は、静岡県・うなぎ・お茶という情報が得られたので、この中から最新の情報を得る。その際、Google Web API を用いて、グーグルニュースから情報を得ることで、最新的话题を得る。今回は「うなぎ」について検索したとする。検索した結果「7月29日は土用の丑の日」だということがわかれば、

**エージェント**「そういえば、7月29日は土用の丑の日だそうです」

と話す。今回は例として、出身地の名産を提示した後に最新の情報を提示したが、本来はランダムで提示しようと考えている。その後、また別のプロフィールを埋めるように、「あなたの好きな食べ物は何ですか？」や「あなたの好きな映画を教えてください」等の別の質問をする。

次にエージェントがはじめに

エージェント「あなたの好きな小説はなんですか？」

という質問をした時の例をあげる。これに対して

ユーザ「坊っちゃん」

と答えたとする。その場合グローバルな情報とローカルな情報に「坊っちゃん」が追加される。この時、エージェントは Wikipedia で「坊っちゃん」を検索する。検索結果から「坊っちゃん」のページを参照し、作者を取得する。作者である「夏目漱石」をグローバル情報に追加し、「夏目漱石」の Wikipedia のページを参照する。このページで作品を取得し、その中からランダムでユーザに提示する。今回は、

エージェント「吾輩は猫である」

と、提示したとする。作品を検索して作者がわからなかった場合、ユーザに対して

エージェント「作者はだれですか？」

と尋ね、その情報を元に、同じ作者の別の作品を検索しユーザに提示する。小説以外にも、好きな音楽や好きな映画に対しても同じアーティスト、同じ監督・俳優を検索してユーザに提示する。

## 6 まとめ

これまでの「サービスマインドを伝えるエージェント・トレーニングシステムの研究」ではエージェントと対話することで接客業の基礎を学ぶことができるシステムを開発した。このシステムでは特定のパターンによってエージェントは特定のパターンで応答するようになっていた。しかし本研究では、必要最低限の知識や情報のみを所有し、その情報を元にユーザとの対話を繰り返し、エージェントが対話の中で成長していくようなシステムの開発を目指す。これにより多くのパターンを予め入力する必要がなくなる。エージェントは初め、ユーザのプロフィールを取得するように質問をし、ある程度情報が集まった場合、保持している情報からユーザの興味があると推測できる最新の情報をユーザに対して提供する。最新の情報を提供する事によって、ユーザを飽きさせないような対話が可能となると考えられる。

## 参考文献

- [1] 守田 航大: サービスマインドを伝えるエージェント・トレーニングシステム, 公立はこだて未来大学卒業論文(2013)
- [2] 守田 航大, 角 薫: 接客トレーニングのためのシリアスゲームの研究, 日本デジタルゲーム学会 2013 年度年次大会, pp.9-13, 日本デジタルゲーム学会 (2014)
- [3] 守田 航大, 角 薫: 接客エージェントによる接客トレーニングシステムの研究, 2014 年度人工知能学会全国大会, 人工知能学会(2014)
- [4] 江端竜次: サービスマインドを伝えるキャラクターエージェント対話システムの研究, 2012 年度公立はこだて未来大学卒業論文, (2013)
- [5] 江端竜次, 角薫: サービスマインドを伝えるキャラクターエージェント対話システム, インタラクション 2013, (1EXB-45), 情報処理学会(2013)
- [6] Kaoru Sumi: Communication with a Virtual Agent via Facial Expressions, IJCAI Workshop on Empathic Computing, 4th International Workshop on Empathetic Computing (IWEC' 13), Beijing, China (2013). (招待講演)
- [7] Kaoru Sumi, Ryuji Ebata: Human Agent Interaction for Learning Service-Minded Communication, iHAI2013, 1st international conference on Human-Agent Interaction, (2013)
- [8] Kaoru Sumi, Ryuji Ebata: A Character Agent System for Promoting Service-Minded Communication, Intelligent Virtual Agents, Lecture Notes in Computer Science, LNAI8108, pp.438, Springer (2013)
- [9] 森下 裕道: 心理接客術 お客様の心を一瞬でギュッとつかむ接し方, ソシム (2009)
- [10] 蓬萊博哉, 灘本明代, 田中克己: “トークショーマタファーによる複数 Web ページの受動的視聴”, 情報処理学会研究報告データベースシステム, 5, p155-p162, (2003)