2003－SLP－47　(5)
2003／7／18

音声対話における擬人化エージェントの利用効果の検討

西本 卓也　　中沢 正幸　　嵯峨山 茂樹

東京大学大学院 情報理工学系研究科

〒113-8656　東京都文京区本郷 7-3-1
E-mail : {nishi, nakazawa, sagayama}@hil.t.u-tokyo.ac.jp

あらまし　擬人化エージェントと人間との音声対話において，視覚的な情報を用いて言語情報の補完を行なうことができれば，効率的で直観的な HMI (Human Machine Interface) の実現に役立つと考えられる．本報告では，音声対話パターン記述言語 VoiceXML で実現されるような音声対話アプリケーションにおいて，擬人化エージェントが視線の移動やジェスチャなどの視覚的な表現を用いて情報提示を行なうことの意義について，予備的な検討を行なう．また，このような視覚的情報の利用効果を評価するために，擬人化音声対話エージェントのツールキット Galatea を拡張し，あらかじめ記述されたシナリオに従って感情表現や身体動作を用いながら音声合成を行うことができ，各種の対話実験に合わせてカスタマイズが容易な実験システムを構築したので報告する．

キーワード　擬人化エージェント，音声対話システム，HMI，VoiceXML，Galatea

The Effectiveness of Using Anthropomorphic Agent in Spoken Dialog Systems

Takuya NISHIMOTO, Masayuki NAKAZAWA and Shigeki SAGAYAMA

Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo, 113-8656 JAPAN
E-mail : {nishi, nakazawa, sagayama}@hil.t.u-tokyo.ac.jp

Abstract　In using spoken dialog systems, visual information with the anthropomorphic agent, such as eye-movement and gestures, can contribute to the effective communications, such as clarifying the ambiguities of verbal information, or showing the change of speaking-turns.

This paper reports the development of our VoiceXML-based spoken dialog agent toolkit, and our plan of experiments which evaluate the effects of verbal communications with visual signs.

Key words　Anthropomorphic agent, Spoken dialog system, Human machine interface, VoiceXML, Galatea

1 はじめに

我々は，GUI (Graphical User Interaface) に限定された従来の HMI (Human Machine Interface) の限界を超える手段として，擬人化音声対話エージェントの利用を目指しており，特に擬人化エージェントによる対話実験を効率的に行うために，VoiceXML [*1] による対話記述を検討している．

知能や感情を持ち自律的に振舞う擬人化エージェントの実現に向けては，様々な研究がなされている．しかし，本研究ではむしろ，効率的で直観的な HMI の実現に貢献するような擬人化エージェントの利用方法に注目する．例えば，目的指向の音声対話アプリケーションを開発する際に，感情表現，視線の移動，ジェスチャなどをどのように用いるか，といった指針を得ることを目指している．

本報告では，擬人化エージェントを用いた音声対話において視覚的な情報が果たし得る役割について検討を行ない，我々の研究の位置付けや狙いを述べる．また，さまざまな実験を効率的に行なうための環境構築の試みとして，擬人化エージェントのツールキット Galatea [1] を用いて入出力モダリティの追加や対話モニタ機能の実現など，さまざまなモジュールの追加について検討する．また，Galatea ツール群のひとつとして実装された VoiceXML 処理系 Galatea DM (Dialog Manager) の実装と，モジュール制御の詳細についても合わせて報告する．

2 視覚的情報による言語情報の補完

2.1 関連研究

人間同士での音声を用いたコミュニケーションにおいては，パラ言語情報（プロソディや感情など），非言語情報（個人性など）が言語情報と同時に伝達されており，人間対機械のコミュニケーションにおいてもこれらの情報の重要性が指摘されているが，従来の検討の多くは，音声に含まれる情報にのみ注目して行なわれてきた．

音声対話システムが言語情報と併用して視覚的な情報を用いることには，システムの状況を明確化し，インタフェースの透過性を改善する機能がある．松坂ら [2] は実空間における対話ロボットを実装し，特にシステムが聞き手である状態でのアウェアネスの改善を実現している．

人間同士が実時間でのコミュニケーションを実現するためには，文字によって伝えられる情報に加えてさまざまな付加情報が必要である [3]．音声においては，声の高さや強さ，ポーズなどのプロソディ情報が情報伝達の効率化に貢献していること，他の実時間的なコミュニケーション手段においても，例えば手話や指点字などにおいてプロソディに相当する情報が存在すること，などが指摘されている．

また，腕や指を自由に動かせる擬人化エージェントを手話の出力に用いる試みもあるが，手話においては手指動作に加えて，非手指動作（表情，口形，うなずき，視線など）が，統語論的・意味論的に重要であるとされる [4]．

2.2 視覚的表現の使用例

我々の目標は，擬人化エージェントが合成音声によって情報を提示する場合に，言語情報を補完し得るような視覚的情報の役割を検討し，有効性を評価することである．具体的には，擬人化エージェントを用いた対話処理系によって，例えば次のような用途での視覚的表現の評価を試みる．

(1) 意味の曖昧性の解消　例えば,「いいです」「けっこうです」といった表現は，文脈によって肯定的にも否定的にも用いられる．文字ではニュアンスを伝えることは難しく，合成音声を用いる場合でも意味を区別して喋ることは難しい．しかし，擬人化エージェントが発話する際に「首を縦に振る動作」「首を横に振る動作」のいずれかを同時に行なえば，合成音声の品質に関わらず的確に意味を伝えられると期待される．

(2) 円滑な話者交替　音声対話システムは，特に音声認識の性能不足を補う必要がある場合に，ユーザに話者交替を適切に行わせるためのガイドが必要となる．電話音声応答システムでは従来，言語音声と効果音を併用して，システム発話の終了やユーザ発話の受理を示す必要があった [5]．一方，人間同士の対面会話においては，話者が話している間は相手から視線をそらし，話者交替のタイミングが近付くと聞き手を見る，といった知見が報告されている [6] ．

発話可能なタイミングをユーザに視覚的に示すことは，多くの音声対話システムで用いられている．特に人間に近い外見や挙動が可能な擬人化エージェントにおいては，人間同士の振舞いを手本にしながら，人間として自然に見えるように，話者交替などを挙動で示すことが重要である．

2.3 対話処理系への要求

本研究における実験ツールとして，我々は，以下の特長を持つ擬人化音声対話エージェントの処理系が必要であると考えている．

1. 多様な視覚的表現が可能であること

- 実写をベースにしてリアルに表情を表現できるエージェントを利用できること．
- 手足を持ちさまざまなジェスチャを行なうことができるエージェントを利用できること．

2. 使いやすい対話マネージャを有すること

[*1] http://www.w3.org/Voice/

- ブートストラップ対話システムを効率的に実装でき，反復的開発が容易であること．
- 実験を監視する第三者のためのログ取得やエージェント操作機能を備えること．
- 複数モダリティを同期的に，あるいは同時に制御できる機能を有すること．

2.4 Galatea ツールキット

前述したシステムを開発するに当たって，我々は Galatea ツールキット（Linux 版）[7] のアーキテクチャを使用する．Galatea は音声認識モジュール (SRM)，音声合成モジュール (SSM)，顔画像生成モジュール (FSM)，エージェント管理部 (AM)，対話管理部 (DM) から構成され，それぞれのモジュールは共通の仮想マシンモデルに基づいて通信を行う．DM は AM をサブプロセスとして起動する．AM は各モジュールを AM のサブプロセスとして起動し，標準入出力を介して通信を行う．

AM は，DCL (Direct Command Layer) および MCL (Macro Control Layer) から構成されている．DCL は各モジュールを直接制御するコマンドセットを提供し，主に DM からみた利便性を実現する．これに対して MCL は，エージェントからの音声出力とリップシンクなどの同期処理を行い，これに必要な状態遷移の管理を行う．また，DM によって定義されたマクロコマンドの処理を行う．さらに，各モジュールが発したメッセージの中で，送信先が指定されていない（ブロードキャスト）メッセージの配送処理などを行なう．

Galatea はカスタマイズが容易で拡張性が高い点が特長とされており，各モジュールは単体での実装，テスト，利用が容易に行える．拡張性や柔軟性に関する検討として，AM を利用する複数の対話管理部（タスクプログラム）の実装が可能であることが確認されている [7] ．

3 モジュールの追加

我々は Galatea に新たな入出力モジュールを追加し，VoiceXML 処理系 (Galatea DM) において対話タスク記述の中にサブモジュール固有の命令を埋め込むことについて検討している [8] ．本章では，我々が行なっているモジュール追加作業について述べる．本報告における Galatea のモジュール構成を表 1 に示す．ただし下線は，IPA の支援を受けて開発されたシステム [1] の最終版から新たに追加・拡張したモジュールである．

3.1 Usherette キャラクタ

我々は，Galatea FSM との互換性を考慮しつつ，身体動作が可能な 3 次元アニメーションキャラクタの表示システム Usherette (図 1) を実装した．実行できる機能は，表情の制御（笑う，驚く，怒る，など），口の制御（「あ」「い」「う」「え」「お」の母音に対応），部分動作（手を握る，指差す，頭を振る，頭を傾ける，など），全体動作（うなずく，お辞儀をする，肩をすくめる，腕を組む，など），自律動作（眼球運動，体の細動）などである．個々のアニメーションはスムーズに連続して行うことができる．

VoiceXML で記述されたエージェント制御シナリオを用いて，Galatea DM から Usherette を制御することが可能である．また，Galatea FSM と Usherette を同時に動かすこともできる．ただし，Usherette 固有の機能については，新たにシナリオに命令を追加する必要がある．

表 1: Galatea のモジュール構成．ただし* はブロードキャスト指定．下線は新たに追加・拡張したもの．

モジュール名	機能
DM	VoiceXML 処理系
AM (AM-DCL)	各モジュールの制御
AM-MCL *	出力同期およびマクロ処理
DM-MCL *	DM 関連のイベント処理
FSM	顔画像合成
SSM	テキスト音声合成
SRM	音声認識
SIM	意味解釈（スケルトン）
GUI	ユーザ向け画面
MON	対話監視者向け画面
SND	音声ファイル出力
PAR	出力の並列化
Usherette	FSM 互換のキャラクタ

図 1: Usherette キャラクタの画面

図 2: ユーザ用画面（ボタン配置は実装の一例）

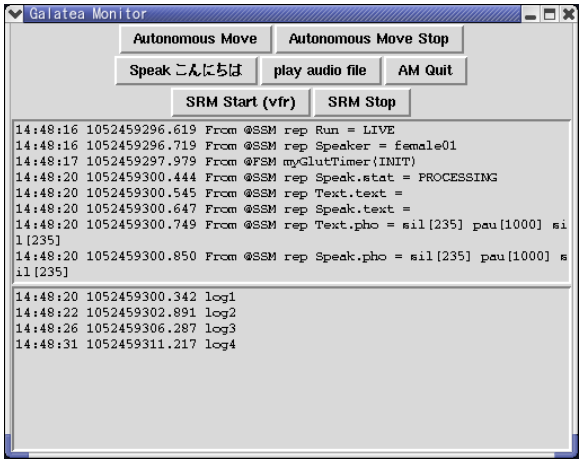

図 3: 実験者用対話モニタ画面の例（上段：ボタン入力，中段：AM のログ表示，下段：DM のログ表示）

3.2 出力手段の追加

擬人化エージェントと人間の対話を検討するための実験には次のような出力手段が必要となる．

(1) 音声合成（リップシンクを含む）

(2) ユーザ向け出力（テキストや画像の表示，オーディオ出力，状況に応じた顔画像の制御など），音声合成と顔画像制御の並列実行

(3) 実験者向けモニタ出力（各サブモジュールのログ出力，対話管理部の内部状態の出力）

これらのうち音声合成と顔画像制御はすでに SSM, FSM によって実現されている．そこで以下のモジュールを新たに実装した．

SND　効果音や録音音声の出力を行う．ファイル名を指定して再生を開始し，出力の開始および終了がモジュールから返答される．

GUI　ユーザ向けの画面表示（図 2）において，テキスト表示やボタン操作による入力機能（後述）を提供する．

MON　実験者向けのモニタ表示（図 3）において，ボタン入力，AM のログ表示，DM のログ表示を行う．

PAR　音声出力と並行して任意のコマンド列を実行する（図 4 参照）．ただし合成音声の特定の箇所とジェスチャを同期させるためには，あらかじめ合成音声の内容に応じてジェスチャのタイミングを指定する必要がある．

SND は Perl 言語により実装され，音声ファイルの再生は外部プログラムの呼び出しによって行う．また，GUI, MON, PAR は Ruby 言語によって実装される．

3.3 入力手段の追加

GUI および MON の各モジュールにおいては，被験者による音声 (SRM) と等価なマウス入力およびマルチモーダル操作，実験者による各モジュールの直接

```
[to @PAR]
set Init = 1
set Cmd = sleep 1.0
set Cmd = to @FSM set FaceExp = HAPPY 1 100 1000
set Cmd = sleep 1.4
set Cmd = to @FSM set FaceExp = SAD 1 100 1000
set Cmd = sleep 2.0
set Cmd = to @FSM set FaceMot = NOD 4

[to @AM-MCL]
set Speak = 私には、うれしいことも、悲しいことも、いろいろあります。
```

図 4: モジュール PAR のコマンド例．「うれしいことも」を発話しながら喜びの表情を，「悲しいことも」を発話しながら悲しみの表情を行ない，最後にうなずく．

制御などを行える機能が望まれる．我々はこれらの機能を，W3C が提案する MMI Framework [9] を参考にしながら実現することにした．

マルチモーダルの入力操作においてはモダリティ統合処理が必要である．また W3C は現在，音声認識文法へのタグの埋め込みによる意味解釈処理について仕様を策定中である． 我々はこれらの機能について，将来的には W3C 準拠も視野に入れつつ，現時点ではスケルトンのみを提供し，実装はアプリケーション開発者に委ねる．

当面必要と考えられる，MON および GUI モジュールからのボタン操作と SRM からの音声認識結果を統合する役割を，新たなモジュール SIM (Semantic Interpretation Module) に持たせた．また，SRM や GUI からのユーザ入力イベントを SIM に転送する役目を持つ DM-MCL モジュールを実装した．

SRM, MON, GUI は各モジュールにおけるユーザ入力イベントを送り先の指定なしに出力する．DM-MCL はブロードキャスト指定モジュールとしてこれらのイベントを監視し，逐次 SIM に送る．SIM は受理したイベントから意味的情報を抽出する（現在の実装では単に認識結果から文字列を抜き出す）．DM は

```
<form id="main">
 <field name="place">
  <prompt> 場所をどうぞ。</prompt>
  <prompt count="3">
   東京と京都のどちらですか？
  </prompt>
  <grammar><rule><one-of>
   <item><token sym="とうきょう">東京</token></item>
   <item><token sym="きょうと">京都</token></item>
  </one-of></rule></grammar>
 </field>
</form>
```

図 5: プロンプト選択を行なう VoiceXML コードの例

ユーザ入力として SIM からのメッセージを監視する．

SIM のスケルトンの実装は Perl で行なうが，使用する文法に応じた拡張や置き換えを想定している．

なお，GUI および MON モジュールにおいて，備えるべきボタンは動的に変更できることが好ましいが，特定のアプリケーションを前提としてボタンを追加修正する場合には，Ruby/TK コードの改変のみで簡単に対応できる．将来はアクティブな音声認識文法に応じてボタンの更新を行なう，などの機能について検討したい．

4 Galatea DM の実装

我々は Galatea のための VoiceXML 処理系 (Galatea DM) を開発している [10] ．使用言語は Java (J2SE 1.4) であり，2.4 節で述べた AM をサブプロセスとして実行する．

VoiceXML はシステム主導型のメニュー選択およびスロットフィリングを行うための制御機能を持つが，変数などのオブジェクトは ECMAScript [*2] の仕様に基づいており，多くの VoiceXML 要素で ECMAScript の実行文や条件式を記述することを許している．我々は ECMAScript エンジンとして Mozilla Rhino [*3] を使用している．

Galatea DM では VoiceXML の制御機能を ECMAScript を含んだ内部命令に変換し，対話状態の遷移に応じて内部命令を実行する．例えば，VoiceXML はユーザ入力が失敗したりユーザが発話しなかった場合にシステムがプロンプトを選択するアルゴリズム[*4] を有するが，このような場合に図 5 の VoiceXML コードは図 6 の中間コードに変換して実行される．内部命令への変換のみを行うことや，内部命令に変換されたシナリオを読み込んで実行することも可能であり，VoiceXML コードが開発者の意図通りに解釈されているか否かをチェックすることが容易になる．

Galatea DM は以下のように出力項目 (OutItem) を分類している（カッコ内は対応する VoiceXML 要素）．

[*2] http://www.ecma-international.org / publications / standards / ECMA-262.HTM

[*3] http://www.mozilla.org/rhino/

[*4] http://www.w3.org/TR/voicexml20/#dml4.1.6

```
<state id="@main.place">
 <cmd>
  <next>'@main.place.noinput'</next>
  <cmd cond="!main.place$.noprompt">
   <script with="main.place$">
    p0=true; p1=true; cc=1;
    if(1<=promptcount && cc<1){cc=1};
    if(3<=promptcount && cc<3){cc=3};
    if(1!=cc){p0=false};
    if(3!=cc){p1=false};
    promptcount++;
   </script>
   <cmd cond="main.place$.p0">
    <add><voice>' 場所をどうぞ。'</voice></add>
   </cmd>
   <cmd cond="main.place$.p1">
    <add>
     <voice>' 東京と京都のどちらですか？'</voice>
    </add>
   </cmd>
  </cmd>
  <add><break length="10.0"/></add>
 </cmd>
 <nomatch next="@main.place.nomatch"/>
 <catch next="@main.place.$1">東京</catch>
 <catch next="@main.place.$2">京都</catch>
</state>
```

図 6: プロンプト選択を行なう中間コードの例

VoiceOutItem　音声合成 (prompt 要素)
AudioOutItem　音声ファイル出力 (audio 要素)
LogOutItem　ログ出力 (log 要素)
NativeOutItem　Galatea 制御コマンド (native 要素)

これらは出力キューによって管理される．VoiceOutItem, AudioOutItem に関しては，出力終了イベントを受け取るまで次の出力を行なわない．NativeOutItem は VoiceXML が定義していない顔画像などの制御や拡張された各モジュールへの出力を行なう．

各 OutItem は出力内容および実行条件を ECMAScript 式として保持し，実際に出力される時点で Rhino によってこれらの式を評価し，その値を用いて出力を実行する．

Galatea DM は以下の手順で出力の処理を行う．

ドキュメント解釈処理　出力に関する VoiceXML 要素を，出力項目 OutItem を出力キューに追加するコマンド (AddOutItem) に変換する．

状態遷移処理　AddOutItem コマンドを実行し，出力キューに OutItem を追加する．

出力キュー処理　OutItem の実行条件を ECMAScript 値として評価し，実行条件が真であれば，出力内容を ECMAScript 値として評価し，評価結果の文字列を出力する．

Galatea 入出力モジュールのコマンドを native 要素として VoiceXML に埋め込んだ例を図 7 および図 8 に示す．

5 まとめ

視覚的情報による言語情報の補完について，擬人化エージェントの利用を前提として予備的な検討を行

```
<block>
 <log>greeting begin</log>
 <native>to @FSM set HeadRotAbs.1 = 0 10 0</native>
 <prompt>こんにちは</prompt>
 <native>to @FSM set HeadRotAbs.1 = 0 0 0</native>
 <log>greeting end</log>
</block>
```

図 7: VoiceXML 拡張によるエージェント制御の例（顔の向きを変えて「こんにちは」と発話し，顔の向きを元に戻す）.

```
<block>
 <native>to @PAR set Init = 1</native>
 <native>to @PAR set Cmd = sleep 0.5</native>
 <native>to @PAR set Cmd = to @FSM set FaceMot = NOD 4</native>
 <prompt>けっこうです。<break/></prompt>
 <native>to @PAR set Init = 1</native>
 <native>to @PAR set Cmd = sleep 0.5</native>
 <native>to @PAR set Cmd = to @FSM set FaceMot = REFUSE 3</native>
 <prompt>けっこうです。<break/></prompt>
</block>
```

図 8: VoiceXML 拡張によるエージェント制御の例（「けっこうです」と発話しながら首を動かす）.

なった．また，Galatea ツールキットの高い拡張性を活かして，視覚的情報の利用効果を評価するための処理系を実装した．シナリオ記述に VoiceXML を用いることで，対話実験に必要なアプリケーション開発が効率化される．我々は Galatea にいくつかのモジュールを追加し, その上で対話マネージャ (Galatea DM) が動作する設計とし，VoiceXML として記述されたシナリオに従って，ジェスチャや表情の制御と音声出力の同時実行，実験者の対話への関与，ログの取得などが容易に実現できることを確認した．

今後は本ツールを用いて被験者実験などを行ない，提案手法の有効性を検討する予定である．また，現在 native 要素によって行なっているエージェント制御については，より簡便な記述方法を検討したい．特に，音声合成 (SSM) から出力されるメッセージの特定の箇所と，特定のジェスチャ命令の同期を行なう，といった処理が容易に記述できることが望まれる．なお，Galatea プロジェクトではツールキットの公開を目指して準備中である[*5] ．

謝辞

本研究の一部は，情報処理技術振興協会 (IPA) および「21 世紀 COE プログラム」の支援を受けた．また VoiceXML 処理系の開発に協力していただいた京都工芸繊維大学の荒木雅弘助教授および岐津三泰氏，Galatea の開発を通じて御議論いただいた Galatea プロジェクトメンバーの皆様，および嵯峨山研究室の皆様に感謝します．

[*5] http://hil.t.u-tokyo.ac.jp/∼galatea/ に情報掲載予定.

参考文献

[1] 嵯峨山茂樹, 他: “擬人化音声対話エージェントツールキット Galatea,” 情処研報 2003-SLP-45-10, pp.57-64, (2003.2).

[2] 松坂要佐, 東條剛史, 小林哲則: グループ会話に参与する対話ロボットの構築, 電子情報通信学会論文誌 Vol. J84-D-II, No. 6, pp. 898-908, (2001.6).

[3] 市川熹: 対話の言葉と障害のある人々, 情報・通信・放送技術におけるユニバーサルデザインに関する国際ワークショップ, pp. 43-54, 東京, (2003.6).

[4] 長嶋祐二, 神田和幸: 手話のコンピュータ処理, 電子情報通信学会誌 Vol. 84, No. 5, pp.320-234, (2001.5).

[5] 高山元希, 西本卓也, 荒木雅弘, 新美康永: 電話音声応答システムにおける効果音の役割, 信学技報 SP 2001-132, pp.55-62, (2002-01).

[6] 前田真季子, 堀内靖雄, 市川熹: 話者交替における視線とうなずきの分析, 人工知能学会研究会資料 SIG-SLUD-A201-09, pp.53-58, (2002.6).

[7] 川本真一, 下平博, 他: “ カスタマイズ性を考慮した擬人化音声対話ソフトウェアツールキットの設計, ” 情報処理学会論文誌, vol.43, no.7, pp.2249-2263, Jul 2002.

[8] 西本卓也, 嵯峨山茂樹: 擬人化エージェント Galatea のための VoiceXML 処理系, 第 17 回人工知能学会全国大会, 2C2-04, (2003.6).

[9] 中村有作, 桂田浩一, 山田博文, 新田恒雄: “ MMI 記述言語の標準化動向と XISL の対応について, ” 信学技報 SP2002-160, pp.73-78, 2002.

[10] 岐津三泰, 西本卓也, 荒木雅弘: 擬人化エージェントのための VoiceXML 処理系の開発, 人工知能学会研究会資料 SIG-SLUD-A201-01, pp.1-6, 2002-06.