音声言語情報処理 45−10
（２００３．２．７）

擬人化音声対話エージェントツールキット Galatea

嵯峨山茂樹 *1 川本真一 *2 下平博 *2 新田恒雄 *3 西本卓也 *1 中村哲 *4 伊藤克亘 *5
森島繁生 *6 四倉達夫 *6 甲斐充彦 *7 李晃伸 *8 山下洋一 *9 小林隆夫 *10
徳田恵一 *11 広瀬啓吉 *1 峯松信明 *1 山田篤 *12 伝康晴 *13 宇津呂武仁 *14

*1 東大, *2 北陸先端大, *3 豊橋技科大, *4 ATR, *5 産総研, *6 成蹊大, *7 静岡大,
*8 奈良先端大, *9 立命館大, *10 東工大, *11 名工大, *12 ASTEM, *13 千葉大 *14 京大

あらまし　筆者らが開発した擬人化音声対話エージェントのツールキット *“Galatea”* についてその概要を述べる。主要な機能は音声認識、音声合成、顔画像合成であり、これらの機能を統合して、対話制御の下で動作させるものである。研究のプラットフォームとして利用されることを想定してカスタマイズ可能性を重視した結果、顔画像が容易に交換可能で、音声合成が話者適応可能で、対話制御の記述変更が容易で、更にこれらの機能モジュール自体を別のモジュールに差し替えることが容易であり、かつ処理ハードウェアの個数に柔軟に対処できるなどの特徴を持つシステムとなった。この成果はソース公開し、一般に無償使用許諾する予定である。

キーワード　擬人化エージェント、音声対話システム、顔画像合成、ソフトウェアツールキット

Galatea: An Anthropomorphic Spoken Dialogue Agent Toolkit

Shigeki Sagayama[*1] *Shin-ichi Kawamoto*[*2] *Hiroshi Shimodaira*[*2] *Tsuneo Nitta*[*3]
Takuya Nishimoto[*1] *Satoshi Nakamura*[*4] *Katsunobu Itou*[*5] *Shigeo Morishima*[*6]
Tatsuo Yotsukura[*6] *Atsuhiko Kai*[*7] *Akinobu Lee*[*8] *Yoichi Yamashita*[*9]
Takao Kobayashi[*10] *Keiichi Tokuda*[*11] *Keikichi Hirose*[*1] *Nobuaki Minematsu*[*1]
Atsushi Yamada[*12] *Yasuharu Den*[*13] *Takehito Utsuro*[*14]

*1 Univ. Tokyo, *2 JAIST, *3 Toyohashi Univ. of Tech., *4 ATR,
*5 AIST, *6 Seikei Univ., *7 Shizuoka Univ., *8 NAIST, *9 Ritsumeikan Univ.,
*10 Tokyo Inst. of Tech., *11 Nagoya Inst. of Tech., *12 ASTEM, *13 Chiba Univ. *14 Kyoto Univ.

Abstract　This paper describes the outline of “Galatea,” a software toolkit of anthropomorphic spoken dialog agent developed by the authors. Major functions such as speech recognition, speech synthesis and face animation generation are integrated and controlled under a dialog control. To emphasize customizability as the dialog research platform, this system features easily replaceable face, speaker-adaptive speech synthesis, easily modification of dialog control script, exchangeable function modules, and multi-processor capability. This toolkit is to be released shortly to prospective users with an open-source and license-free policy.

Keyword　anthropomorphic agent, spoken dialog system, animated face image, software toolkit

1　はじめに

著者らは、擬人化音声対話エージェントのソフトウェアツールキット “Galatea[1]” を開発し、近々オープンソース、ライセンスフリーのソフトウェアツールキットとして公開すべく準備中である。

この計画の源流は、1994 年の情報処理学会 音声言語情報処理研究会の発足時に行われた「なぜ音声認識は使われないか」と題する議論である。この中で、音声認識性能の向上だけでは音声認識技術利用が進まないのではないかとするさまざまな意見が出された。その後、同研究会の「マルチモーダルツールワーキンググループ」(1998-2000) で、今後の音声研究者の研究目標を議論し、次世代の研究ターゲットとして擬人化エージェントを構想し、その研究プラットフォームを研究者の共同作業により構築して

[1]ガラテア (Galatea) はギリシア神話に登場する女性。キプロスの王ピグマリオン (Pygmalion) は大理石で理想の女の像を彫り上げ、それに恋をしてしまった。美の女神アフロディテは彼の願いを聞き入れ像に命を与えた。女はガラテアと呼ばれ、ピグマリオンと結婚し、息子パフォスをもうけ幸福に暮らした。この話は、ミュージカル “My Fair Lady” の素材ともなった。我々自身を現代のピグマリオンに模し、システムが真に人間的になるよう願ってシステムをこのように命名した。

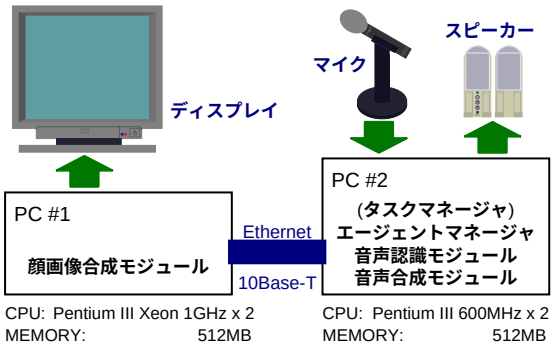

図 1: Galatea システムの動作環境例

公開する計画を持った。この構想は、2000〜2002 年度に情報処理技術振興協会 (IPA) の支援を受け、主に大学の十数名の研究者が協力体制を作って実行した [1]。

近年、音声対話エージェントの研究が盛んになって来ており、ツールキットの開発もなされている [2, 3, 4, 5, 6] が、多くはコンピュータグラフィクスで顔画像を合成し、手作業で組み立てた仮想人物であり、いわばゲームソフトの 1 キャラクタにのように、その容貌や合成音声を変えることは容易でない場合が多い。また、音声認識、音声合成、顔画像生成などのモジュールまで含み、それらをオープンソースで、無償使用できるツールは、少なくとも日本語についてはまだない。このような途上の技術を用いてさまざまな音声言語対話の用途を開発するためには、カスタマイズ可能性やソースレベルでの変更可能性が重要である。これが本ツールキットを開発した主要な動機である。

そもそも音声認識や音声合成の技術開発は、「人間と機械が対話をする夢の技術」を実現するためである、としばしば言われて来た。しかし、個別の技術の研究は盛んになされたが、これらを統合して「夢」の実現に近付く研究活動は、相対的に不足している。その結果、個別の技術の研究は統合を必ずしも意識しないで進められ、統合に関する問題は先送りされて来たと思われる。統合の研究のためのプラットフォームが整備され使いやすくなれば、対話に用いる個別技術の新たな問題も発掘することができ、研究に資することができると期待される。

2 Galatea の全体構成と特徴

2.1 本ツールキットの特徴と全体構成

本ツールキットは、音声認識、音声合成、顔画像合成の 3 基本機能を統合し、対話制御のもとでユーザと対話するエージェント、およびその開発環境を提供するものである。想定する機器構成を図 1 に示す。特徴としては

図 2: Galatea システムの画面表示例

図 3: Galatea システムとの対話風景

- 高いカスタマイズ可能性 (顔、合成音声、認識文法、対話制御など)。
- 標準化動向に対応 (VoiceXML, W3C, JEIDA-62-2000 など)。
- 簡明なモジュール間通信。部品交換が容易。モジュール別に別々の PC に分散して実行可能。
- 最新の高度な技術内容を実現。特に、初の無償の日本語テキスト音声合成システムが含まれている。
- ソース公開、無償使用許諾を予定。

などを挙げることができる。

図 1 に示す分散環境、あるいは Mobile Pentium III 1.2GHz の CPU と 512MB のメモリを搭載したノート PC 1 台でも動作が確認されている。システム画面、およびシステムとの対話風景を図 2、および図 3 に示す。

全体構成を図 4 に示す。基本的な構成では、

- 対話音声認識モジュール (SRM)
- 対話音声合成モジュール (SSM)
- 顔画像合成モジュール (FSM)

の 3 機能モジュールをモジュール統合処理部 (Agent Manager: AM) が統合し、タスク制御モジュール

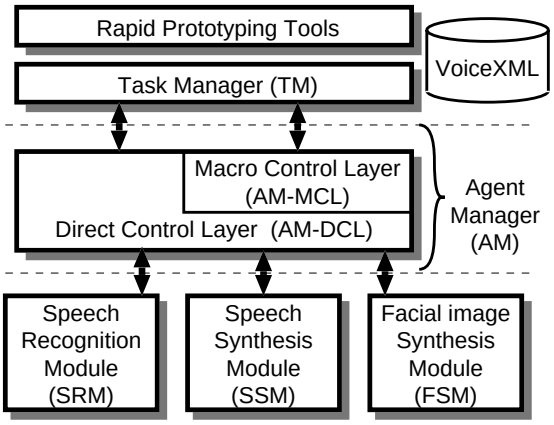

図 4: AM と各モジュールとの接続関係

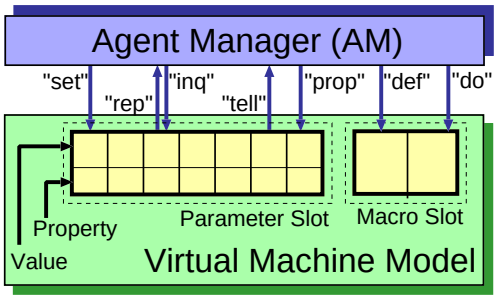

図 5: AM と仮想マシンモデルとの関係図

(TM) あるいは対話制御モジュール (DM) の下で動作する。モジュールを追加することは容易なので、実際に頭部動作制御モジュールの追加も試みた。

各モジュールは独立したプロセスとして、単一の PC、もしくは複数の PC 上で並行に動作することを想定している。モジュール統合処理部は各モジュールが連動して 1 つの対話システムとして円滑に動作するためのシステム制御、情報管理などを司る。

2.2 モジュール統合の方式

AM と各モジュールとの接続関係は図 4 に示すように、大きく分けて 2 つの機能レイヤーで構成される。Direct Control Layer (AM-DCL) は、各モジュールの規定するコマンドセットを直接制御する事を可能とするレイヤーであり、多くのモジュールはこのレイヤーを介して他のモジュールとの通信を行う。

Macro Control Layer (AM-MCL) は、主に対話タスク記述などを管理するタスク管理部 (TM) 向けのレイヤーで、良く使われる一連のコマンドセットをまとめたマクロコマンドとして再定義したり、モジュール間の同期管理などの低レベルなモジュール制御を請け負うことで、TM から見た利便性を向上させる。例えば、エージェントの音声発話時における合成音声と合成画像中の口の運動の同期 (以後 Lip Sync と呼ぶ) は、マクロコマンドとして実現した。

表 1: 通信コマンド名とその機能

名前	機能
set	パラメータスロットの値の設定
def	マクロスロットの値の設定
inq	スロットの値の問い合わせ
prop	スロットの属性の設定
save	現在のスロットの値を別名をつけて蓄積
rest	save で蓄積されたスロットの値の復帰
del	save で蓄積されたスロットの値の解放
do	マクロスロットの値 もしくはファイルの内容の評価

簡明で実現が容易なモジュール間の通信方式として、次のような方式を採用した。

- 各モジュールは、UNIX の標準入出力を通して通信する。モジュール間の通信は、必ず AM を介して行なう。
- 通信はコマンド形式で行う。コマンドの種類を表 1 に示す。
- 各モジュールは、機能をスロットとして定義した仮想マシンモデル (図 5) として統一したインタフェースにて扱う。

これにより、モジュールの分散並行処理ができ、モジュール追加などが容易になった。また、モジュールの単体利用・開発・試験を容易になった。

3 音声認識モジュール

3.1 音声認識モジュールの機能・仕様

音声対話システムのための音声認識システムには、認識性能が高精度かつ高速であるだけでなく、探索用パラメータの制御、認識処理の中断・再開、結果の逐次出力、複数文法の切り替え、および出力形式の選択などの様々な機能が求められる。このモジュールでは基本的に以下の機能を実現する。

- 文法に基づく音声認識
- 発話中の逐次的な認識結果出力
- 認識処理の動的な制御（中断、文法切り替えなど）

音声認識モジュールの構成を図 6 に示す。音声認識モジュールは音声認識エンジン、通信・制御モジュール、文法変換モジュールの 3 つのサブモジュールからなる。

図のように音声認識実行部と通信・制御部を独立させることで、外部プログラムと音声入力から非同

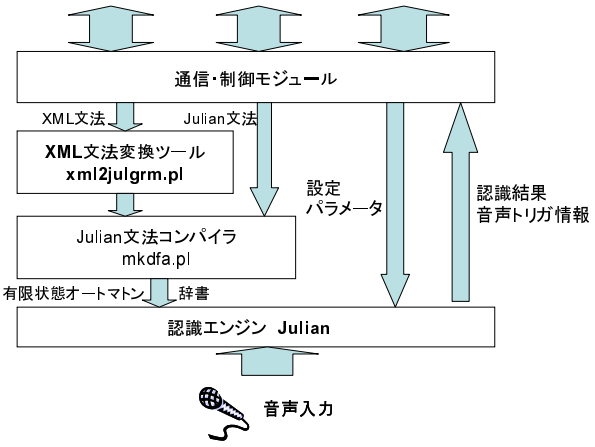

図 6: 音声認識プログラムの構成

期に発生する通信イベントと音声入力イベントに対してそれぞれ専属のプロセスを割り当て、イベントの取りこぼしや遅延を防ぐ設計となっている。外部通信・制御部は、プログラムからのコマンドの受付・スロット値の管理やマクロ処理・認識結果の再フォーマットなどの通信部分、および認識エンジンの起動や中断・文法の流し込みなどのエンジン制御を行う。

モジュールの中心となる音声認識部は、認識対象単語をその発音のサブワード列とともに記述した単語辞書、それらの構文上の接続制約を記述したタスク文法、各単語の音響尤度を与えるサブワード単位の HMM 音響モデル、および認識エンジン本体から構成される。音声認識エンジンは Julian[7] を想定しているが、後述の実行例に示すように文法や出力形式のインタフェースを汎用的なものとし、他の認識エンジンに置き換えても外部モジュールからは等価に扱える仕組みになっている。

認識対象とする発話の語彙や構文規則は外部モジュールから与えられる。Julian はオートマトン文法のみを扱うので、文法は専用コンパイラによって有限状態オートマトン（FA）に変換される。音響モデルはサブワード単位の HMM を用いる。ファイルのフォーマットは標準的な HTK[8] のフォーマットに対応する。

3.2 逐次出力・文法切り替え

認識結果の出力は、一位候補だけでなく N-best 解も出力する。出力のタイミングは、一入力区間の認識結果をまとめて出力することを基本とする。また、Julian では探索開始時から入力の進行に沿って、漸次的に 1 パス目の中間結果を出力することもできる。各認識結果の文候補は外部モジュールに出力されるが、認識結果の多様な利用法を想定して XML 形式とし、単純な単語列の他に、時間情報、各単語の文法カテゴリ番号や音素列、各単語がマッチした音声区間のフレーム数、平均音響尤度などの情報を併せて出力する。また、音声入力（認識）中や待機中などの音声入力の状態についても、他のモジュールからの要求に応じて出力する。

文法切替えは、通信・制御モジュールを経由して随時行うことができる。認識処理を行っている間は、送られてきた文法はすべて順にキューに入れられて認識が続行され、文法の切り替えタイミングは音声入力の切れ目ごととなる。上述のような逐次出力、出力形式や文法の切替えなどは、全て外部インタフェースを通じて随時設定を行うことができる。

3.3 文法記述・文法変換ツール

文法仕様としては Julian 形式と XML 形式の 2 種類の文法・辞書の記述形式に対応する。本モジュールで新たに導入した XML 形式による文法・辞書の記述仕様は、W3C による Speech Recognition Grammar Specification の仕様案を参考にしており、基本は文献 [9] における XML 形式の文法の仕様に準じている。すなわち、文法・辞書の記述は主に “トークン (token)” 及び “書き換え規則 (rule)” の並びの定義及び参照からなる。単語の読み情報を付与するため、token タグに phoneme 及び syllable の属性を独自に拡張している。以下に記述の一部の例を示す。

```
<rule id="siritai">
<one-of>
<item><token phoneme="sh;i;r;i;t;a;i;">
  知りたい</token></item>
<item><token phoneme="k;i;k;i;t;a;i;">
  聞きたい</token></item>
</one-of>
</rule>
```

文法変換ツールは、外部から与えられた文法を認識エンジンで使用可能な形式に変換する。このツールは XML の変換技術である XSLT(XML Stylesheet Language Transformation) を応用しており、変換規則のみ変更することで他の認識エンジン（SPOJUS[10]）の形式文法への変換が可能である。

3.4 動作例

以下に実行例の一部を示す（[] で囲まれた行は外部プログラムから入力されたコマンド）。

```
[set Grammar = GramJulian/attendant/name.dfa]
[set Dic = GramJulian/attendant/name.dict]
[set Run = START]
tell <INPUT STATUS="LISTEN" TIME="994678530"/>
発話「小泉さんお願いします」
tell <INPUT STATUS="STARTREC" TIME="994678547"/>
tell <INPUT STATUS="ENDREC" TIME="994678549"/>
```

```
tell << EOM
<RECOGOUT>
  <SHYPO RANK="1" SCORE="-4799.441895">
    <WHYPO WORD="silB" CLASSID="4" PHONE="silB"/>
    <WHYPO WORD="小泉" CLASSID="0"
      PHONE="k o i z u m i"/>
    <WHYPO WORD="さんお願いします" CLASSID="1"
      PHONE="s a N o n e g a i sh i m a s u"/>
    <WHYPO WORD="silE" CLASSID="5" PHONE="silE"/>
  </SHYPO>
</RECOGOUT>
```

4 対話音声合成モジュール

4.1 機能と動作原理

Galatea における対話音声合成モジュール (GalateaTalk) は、漢字仮名混じり文で表記された日本語テキストを合成音声に変換する、いわゆる日本語テキスト音声合成を行う基本的な機能

(1) 形態素解析
(2) 読み、アクセント型情報の付与
(3) 韻律生成
(4) 合成波形生成
(5) 合成音声出力

に加えて、顔画像生成をともなう音声対話システムを構成するための音声合成モジュールとして、以下の機能

(6) 出力発話 (合成音声) における各音素の継続時間長の出力
(7) 埋め込みタグによる韻律の制御
(8) 音声出力の途中停止、および中断における既出力音素列の出力

を持つ。(6) は顔画像出力における口唇の動きと合成音声を同期させるために用いられる。

(1), (2) では、アクセント情報を付加した辞書を用いて “茶筌 [11]” で形態素解析したのち、先行研究 [12] で示されるアクセント処理を行う。(3), (4) では、HMM に基づいた音声合成 [13, 14, 15] により、合成波形を生成する。音声合成部で必要となる話者の音響モデルとしては、男女各 1 名の基本話者のモデルが提供される。

GalateaTalk は、独立した四つのモジュール、コマンド解析部、テキスト解析部、音声合成部、音声出力部から成っており、図 7 の構成をとる。

4.2 入力コマンド

GalateaTalk は、標準入出力を通じたコマンドによって外部と通信する。コマンドは、set, inq, prop の三つで基本的に構成され、それぞれ各種スロット値の設定、問い合わせ、属性変更を行う。例えば、

```
inq SpeakerSet
```

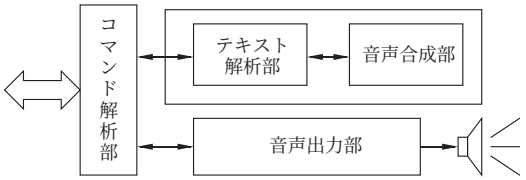

図 7: GalateaTalk の構成

```
<SPEECH> <VOICE OPTIONAL="male1">
これは<PRON SYM="アイピーエー">IPA</PRON>
のプロジェクトで開発された<EMPH>対話</EMPH>
音声合成システムです。
</VOICE> </SPEECH>
```

図 8: JEIDA-62-2000 による発話文の記述例

では、利用可能は話者の情報が標準出力に出力される。また、

```
set Text = こんにちは。
```

では、「こんにちは。」の音声が合成され、他モジュールと同期をとるために、発話文中の音素系列が継続時間長とともに標準出力に出力される。さらに、

```
set Speak = NOW
```

によって、合成音の出力が直ちに行われ、合成音の出力中に

```
set Speak = STOP
```

のコマンドを受け取ると音声出力を停止し、既に音声出力された音素列を出力する。

GalateaTalk では、音声出力する発話文の内容は set コマンドで Text スロットの値を設定することによって行う。発話文の表現形式としては、プレインテキストによる漢字仮名混じり文に加えて、4.1 節 (7) の機能を実現するために、(社) 日本電子工業振興協会「日本語テキスト音声合成用記号の規格 (JEIDA-62-2000)」[16, 17] におけるテキスト埋め込み制御タグおよび仮名レベルの韻律記号に準拠したタグ付きテキストを受け付ける [18]。この記述例を図 8 に示す。

5 顔画像合成モジュール

5.1 機能と原理

Galatea の顔画像合成の基盤として用いたのは IPA プロジェクト「感性擬人化エージェントのための顔情報処理システムの開発」(1995.6〜1998.3) で著者らが開発したソフトウェア [19] である。このソフトウェアは無償公開されており、正面方向から撮影した 1 枚の顔画像と標準顔モデルを整合させ、各個人のモデルを生成できるため、顔画像を準備する

図 9: 表情合成例

図 10: 口画像合成の例

だけでエージェントの顔をカスタマイズできる。今回新たに、より人間らしい対話を実現するために、精密な Lip Sync のための他のモジュールとの連係、喜びや怒りを表現するための任意の表情付加機能、自然な瞬きの制御機能を付加した。

標準顔モデルはワイヤフレームモデルと呼ばれる多面体近似モデルである。この 3 次元モデルは約 800 ポリゴンで形成される。 顔画像と標準顔モデルとを整合する作業は GUI ツールにより行う。

人間の顔表情を画面内のモデルに表現させるためには顔の各部分の動きを定量的に与える表情記述規則が必要である。顔の表情変化を表現する方法として FACS(Facial Action Coding System) を導入している。 FACS は解剖学的に分類された 44 種類の運動単位 AU(Action Unit) から成り立つ。この AU の移動量および移動方向をパラメータとして 3 次元モデルを変形させ表情合成を行う。表情変化は 3 次元モデルの各格子点を AU の強さによって移動させる。図 9 に感情合成結果の一例を示す。

発話時の口の形状を規定する口領域の変形パラメータ (以下、口形パラメータ) を表現するためには、AU とは異なる口領域の変形に限定したパラメータを用いる。これらの口形状を決定するため、口形編集ツールを用いてカスタマイズを行う。口形状は基本的に 13 個の唇の厚みと形状を表現するパラメータ (Viseme) によって記述される。 図 10 に典型的な母音の口形を示す。唇は厚みを持ち、さらに先述した口内のモデルを持っているため、リアルかつ微妙な口の形状表現が可能となっている。

5.2 顔画像生成モジュール

先述した表情・口形パラメータを用いてエージェントの顔モデルの制御を行う。

エージェントに対してアニメーションさせたい表情・口形パラメータは基本的に統合・制御モジュールから送信されてくる。このパラメータを用いて正確に表情をアニメーションさせる。表情に関して、現状のモジュールでは表情の強度および表情の継続時間が処理可能となっている。表情の種類は怒り、喜び、悲しみ、嫌悪、恐れ、驚きの基本 6 感情が操作でき、これらの定義された AU パラメータのデータは予め用意しておく。口形状に関しても同様に統合・制御モジュールから 1 文章分の音素（口形）パラメータと音素長が送信されるが、注意すべき点として合成音声との同期を考慮に入れる必要がある。この問題を解決するために統合・制御モジュールが音声合成、顔画像合成各モジュールに対し同時刻に相対時間での発話開始時間を送信することで解決している。 口形状のリアルなアニメーションを表現させるため、受信した口形パラメータとその継続長に基づいて、キーフレーム位置の配置を行う。これらキーフレーム間を線形補間することで滑らかなアニメーションを生成させることが可能となる。

本モジュールでは統合・制御モジュールから送信されたパラメータによって頭部の制御が可能となっている。また瞬きも制御でき、より自然なエージェント構築が可能となっている。エージェントのキャラクタ変更・追加に関しては登場させたいエージェントの数だけデータを製作し、モジュール起動前にそれらデータを読み込み、要求に応じて切り替えを行う。データ生成は先述した顔モデル生成の GUI ツールを用いることで簡単に構築可能である。

6 MMI 対話記述とシステム開発支援ツール

6.1 基本動作サンプルタスクシステム

このキットには、エージェントの音声発話時における音声と顔画像の同期を中心に、音声認識モジュール (SRM)、音声合成モジュール (SSM)、顔画像合成モジュール (FSM)、およびエージェント管理部 (AM) の基本機能の動作確認のために、簡単な対話

タスクを数サンプル添付する。いずれのタスクも単語数 100 以下、単語パープレキシティ10 以下の非常に簡単なタスクであるが、VoiceXML による対話制御を用いずに統合動作を直接制御する場合のサンプルとして利用できる。

6.2 VoiceXML による対話記述

可読性の高い記述言語を用いた対話制御機能を提供することで、開発者は容易にシステムを用いて対話システムを試作できるようになる。我々は VoiceXML を記述言語として用いた対話マネジャ Galatea DM を実装した [20, 21]。

Galatea DM は、Galatea システムがさまざまな研究やアプリケーションと開発に用いられることを想定して、さまざまな拡張が容易に行えるような設計を行っており、VoiceXML 以外の対話記述言語を容易に追加実装可能である。また、ECMAScript[22] 記述の埋め込みにも対応している。

図 11 に Galatea DM が実行可能な VoiceXML 記述の例を示す。Galatea DM は汎用的な状態遷移モデルを用いており、読み込まれた VoiceXML ドキュメントを独自の内部表現データに変換する。ドキュメント中の音声認識文法 (W3C 準拠の XML 形式) は音声認識モジュール (SRM) に渡されて内部形式に変換される。その後、ECMAScript による値の解釈を行ないながら内部表現データを逐次実行し、対話を実行する。感情を指定した顔制御や音声合成の制御を埋め込んだドキュメントにも対応している。

Galatea DM の実装には Java 言語 (Java2 Standard Edition Version 1.4)、スキーマコンパイラ Relaxer [2]、Mozilla Rhino [3] などを使用し、AM をサブプロセスとして呼び出すことによって各モジュールを制御している。

6.3 プロトタイピングシステム [23]

本システム (Galatea Prototyping System) は、プロトタイピングツールと実行環境から成る。PC 上で動作し、入力モダリティとして、音声のほかマウスとキーボードを、また出力モダリティとして、音声 (TTS) 、顔画像 (表情、リップシンク) のほかウィンドゥへのコンテンツ表示を使用することができる。プロジェクトでは、各モジュールが分散した環境下でも動作するよう、エージェントマネジャが用意されているが、本システム中の実行環境では、同一システム上にすべてのモジュールを置き、Windows 上で動作することを前提に開発している。

[2]http://www.relaxer.org/
[3]http://www.mozilla.org/rhino/

```
<form id="weather">
 <block> 天気情報サービスへようこそ。</block>
 <field name="place">
  <prompt> 場所はどこですか? </prompt>
  <grammar>
  <rule><token phoneme="t;o:;ky;o:;">
   東京</token></rule>
  <rule><token phoneme="k;a;n;a;g;a;w;a;">
   神奈川</token></rule>
  <rule><token phoneme="ch;i;b;a;">
   千葉</token></rule>
  </grammar>
 </field>
 <field name="when">
  <prompt> いつの天気ですか？</prompt>
  <grammar>
  <rule><token phoneme="ky;o;u;">
   今日</token></rule>
  <rule><token phoneme="a;sh;i;t;a;">
   明日</token></rule>
  </grammar>
 </field>
 <block>
  <prompt> <value expr="place"/> の
   <value expr="when"/> の天気です。</prompt>
 </block>
</form>
```

図 11: Galatea DM に対応した VoiceXML の例

図 12 にシステムの持つ実効環境の構成を示す。システムは、フロントエンド、対話制御部、及びドキュメントサーバから構成される。ドキュメントサーバには、システムのドキュメント（対話シナリオ（XISL[24])、データ（XML)、表示スタイル (XSL)）が保持されている。対話制御部は、対話シナリオの解釈・実行、フロントエンドからの入力情報の処理、フロントエンドへの出力命令の送信を行なう。フロントエンドは、Galatea プロジェクトで開発中の各モジュールを利用した入出力インタフェースを持つ。入力インタフェースがユーザからの入力を受け付けると、その内容を対話制御部に送信し、一方、対話制御部からの出力命令を受けると、出力インタフェースがユーザへの出力を行なう。

プロトタイピングツールは、オンラインショッピングと航空券予約の二つのドメインを対象に、MMI 開発支援環境を提供する。図 13 に実行画面の例を示す。開発者は、対話シナリオを作成するための部品 (対話の枠組み、組み合わせ、対話の遷移等) を並べたツールバーと、各モジュールの機能属性を指定する dialog box を使用してアプリケーションを作成することができる（詳しくは文献 [23] を参照)。なお、データと表示に関する部分は、サンプルを参考に XML ドキュメントを予め作成しておく必要がある。

プロトタイピングツールを用いて生成した XISL

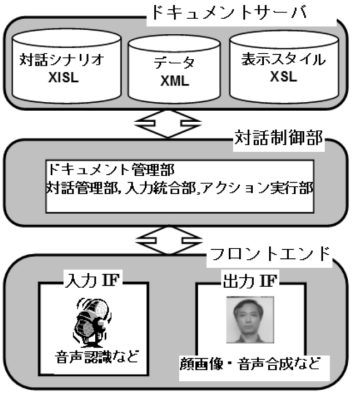

図 12: Galatea Prototyping System の実行環境

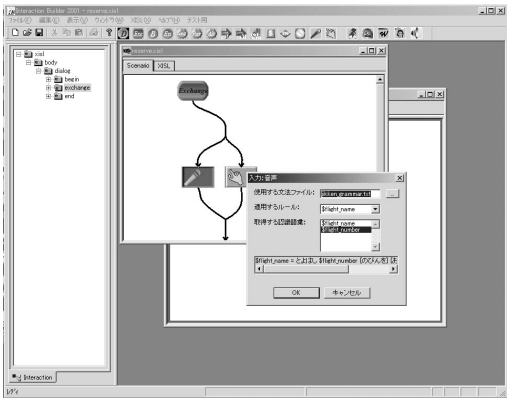

図 13: Galatea Prototyping System の実行画面例

ドキュメントは、上に述べた実行環境を利用して動作テストを行うことができる。

7 おわりに

著者らが開発した擬人化音声対話エージェント開発ツールキット “Galatea” の概要を述べた。今後は

`http://hil.t.u-tokyo.ac.jp/~galatea/`

にダウンロード方法、開発状況などの情報を掲載する予定である。

本成果は、可能な限りソース公開、商業利用も含めた無償使用を認める方向で、使用条件の詳細を検討中である。最終的には、配布キットに含まれる該当文書を参照されたい。

参考文献

[1] 嵯峨山 他: “擬人化音声対話エージェント開発とその意義,” 情報処理学会研究報告 2000-SLP-33-1, Oct. 2000.

[2] D. W. Massaro, M. M. Cohen, J. Beskow and R. A. Cole, “Developing and evaluating conversational agents,” in J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.) *Embodied conversational agents,* Cambridge, MA: MIT Press, 2000.

[3] J. Gustafson, N. Lindberg and M. Lundeberg: “The August Spoken Dialogue System,” Proc. of Eurospeech99, pp.1151-1154, 1999.

[4] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid and V. Zue: “GALAXY-II: A Referece Architecture for Conversational System Development,” Proc. ICSLP98, pp.931–934, 1998.

[5] 土肥, 石塚: “Face-to-face 型擬人化エージェント・インタフェースの構築,” 情報処理学会論文誌, Vol.40, No.2, pp.547-555, Feb. 1999.

[6] 向井, 関, 中沢, 綿貫, 三吉: “非言語情報を用いたマルチモーダル対話インタフェースの試作,” Interaction2001, pp.139-140, 2001.

[7] 李, 河原, 堂下, “文法カテゴリ対制約を用いた A*探索に基づく大語彙連続音声認識パーザ,” 情報処理学会論文誌, Vol.40, No.4, pp.1374–1382 (1999).

[8] S. Young, J. Jansen and J. Odell, D. Ollason P. Woodland, *The HTK BOOK*, 1995.

[9] Speech Recognition Grammar Specification for the W3C Speech Interface Framework - W3C Working Draft 20 August 2001, http://www.w3.org/TR/2001/WD-speech-grammar-20010820/.

[10] 甲斐, 中川, “冗長語・言い直し等を含む発話のための未知語処理を用いた音声認識システムの比較評価,” 電子情報通信学会論文誌, Vol. J80-D-II, No. 10, pp. 2615–2625, 1997.

[11] http://chasen.aist-nara.ac.jp/

[12] 匂坂, 佐藤: “日本語単語連鎖のアクセント規則,” 信学論, **J66-D**, 7, pp. 849–856, 1983.

[13] 益子, 他: “動的特徴を用いた HMM に基づく音声合成,” 信学論, **J79-D-II**, 12, pp. 2184–2190, 1996.

[14] 益子, 他: “多空間確率分布 HMM によるピッチパタン生成,” 信学論, **J83-D-II**, 7, pp. 1600–1609, 2000.

[15] http://hts.ics.nitech.ac.jp/

[16] 赤羽, 蓑輪, 板橋: “音声合成用記号の標準化について,” 音響誌, 57, 12, pp. 776-782, 2001.

[17] (社) 日本電子工業振興協会: 日本語テキスト音声合成用記号の規格, JEIDA-62-2000, 2000.

[18] 山下, 他: “マルチモーダルコミュニケーションのための音声合成プラットホーム,” 情報処理学会研究報告, SLP-40-12, pp. 67-72, 2002.

[19] 森島, 八木, 金子, 原島, 谷内田, 原: “顔の認識・合成のための標準ソフトウェアの開発,” 電子情報通信学会技術報告, PRMU97-282, Mar. 1998.

[20] 西本 他: “擬人化音声対話エージェントのためのタスク管理機能,” 日本音響学会 2002 年春季研究発表会, 1-5-15, pp.29-30, 2002.

[21] 岐津 他: “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 SIG-SLUD-A201-01, pp.1-6, 2002.

[22] http://www.ecma.ch/ecma1/STAND/ECMA-262.HTM

[23] 足立 他: “MMI システム構築のためのプロトタイピングツールの開発,” 情処研究報告 SLP-43-2、pp.7-12 (2001).

[24] http://www.vox.tutkie.tut.ac.jp/XISL/