2008－SLP－73 (9)
2008／10／24

音声対話技術コンソーシアム (ISTC) の活動成果報告

山下洋一(*1), 李晃伸(*2), 河原達也(*3), 四倉達夫(*4), 西本卓也(*5), 桂田浩一(*6), 新田恒雄(*6)

1) 立命館大学, 〒 525-8577 滋賀県草津市野路東 1-1-1
2) 名古屋工業大学, 〒 466-8555 愛知県名古屋市昭和区御器所町
3) 京都大学, 〒 606-8501 京都府京都市左京区吉田本町
4) ATR 音声言語コミュニケーション研究所, 〒 619-0288 京都府相楽郡精華町光台二丁目 2-2
5) 東京大学, 〒 113-8656 東京都文京区本郷 7-3-1
6) 豊橋技術科学大学, 〒 441-8580 豊橋市天伯町雲雀ケ丘 1-1

あらまし 音声対話技術コンソーシアム (ISTC) では，音声対話システムにおけるインタフェース部を容易に構築できるようにするために，音声認識，音声合成，顔画像合成，対話制御の要素技術から構成されるツールキットの開発を進めてきた．本報告では，各要素技術における機能を中心に，ISTC のこれまでの成果を紹介する．

キーワード：音声対話システム，擬人化エージェント，ヒューマンインタフェース，音声認識，音声合成，顔画像合成

Activity Report of Interactive Speech Technology Consortium(ISTC)

Yoichi Yamashita(*1), Akinobu Lee(*2), Tatsuya Kawahara(*3), Tatsuo Yotsukura(*4), Takuya Nishimoto(*5), Kouichi Katsurada(*6), Tsuneo Nitta(*6)

1) Ritsumeikan University, 1-1-1 Nojihigashi, Kusatasu-shi, Shiga, 525-8655
2) Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya-shi, Aichi, 466-8555
3) Kyoto University, Yoshida-Honmachi, Sakyou-ku, Kyoto-shi, Kyoto, 606-8501
4) ATR-SLC, 2-2-2 Hikaridai, Seika-cho, Souraku-gun, Kyoto, 619-0288
5) The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656
6) Toyohashi University of Technology, 1-1 Hibarigaoka, Tenpaku-cho, Toyohashi-shi, Aichi, 441-8580

abstract The Interactive Speech Technology Consortium(ISTC) has been developing a toolkit which is composed of four fundamental modules of speech recognition, speech synthesis, face synthesis, and dialog control, in order to facilitate realizing interface for spoken dialog systems with an anthropomorphic agent. This report describes the outcome of ISTC focusing the function of each module.

keywords: spoken dialog system, anthropomorphic agent, human interface, speech recognition, speech synthesis, face synthesis

1 はじめに

様々な情報機器の開発やロボット技術の進展に伴い，我々が情報交換を行う対象が多様化している．このような多様な機械と容易に情報交換を行うには，従来のキーボードやマウス，あるいはペンによるタッチ入力とディスプレイ出力に基づくインタフェースでは不十分であり，誰でも簡単で使いやすいインタフェースの構築が必要となっている．中でも，音声認識や音声合成をうまく組み入れたインタフェース技術は，使いやすいサービスを提供する上での重要な技術となることが期待される．しかし，音声認識や音声合成は，音声データ収録，モデル学習，システム実装など開発に非常にコストを要する要素技術であり，それを自ら実現することは問題解決を行う個々のアプリケーション開発者にとって容易なことではない．

音声対話を実現するために不可欠な広範囲の基本ソフトウエアを提供することを目的として，2003 年度 11 月に情報処理学会音声言語情報処理研究会のもと，音声対話技術コンソーシアム (ISTC) が設置され活動を行ってきた [1]．本コンソーシアムでは，擬人化エージェントとの音声対話を実現するためのツール

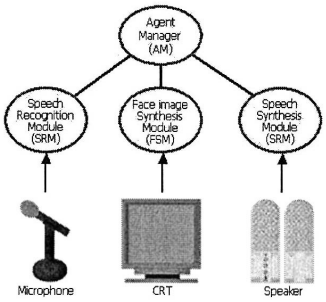

図 1: Galatea ツールキットの基本構成

キット Galatea の開発を行うとともに，マルチモーダル対話 (MMI) 記述の標準化についても検討を行ってきた．本報告では，これまでに得られた ISTC の成果について紹介する．

2 Galatea の構成と機能

2.1 基本構成

Galatea は，擬人化エージェントとの音声対話を行うためのインタフェースを実現するためのツールキットであり，図 1 に示すように，音声認識 (SRM: Speech Recognition Module)，音声合成 (SSM: Speech Synthesis Module)，顔画像合成 (FSM: Face Synthesis Module)，対話制御 (AM: Agent Manager) の 4 つの基本モジュールから構成される．以下，各モジュールの機能について述べる．

2.2 音声認識モジュール: SRM

Galatea の音声認識モジュール SRM は，音声認識エンジン Julius をコアエンジンとして，それに Galatea 用の通信プロトコルを実装したラッパーをかぶせることで実装されている．モジュールとの接続や通信はラッパーが執り行い，必要に応じて Julius のパラメータ設定，起動と停止，動作の制御などを行う．子プロセスとして起動される Julius が認識した結果は，ラッパーを通じて各モジュールに送信される．

本プロジェクトでは，音声認識モジュール SRM および Julius の最新版を開発・公開してきた．Julius は音声対話システム向けに機能，性能および利便性を大幅に向上させてきた．また，Windows Speech API (SAPI) 版 Julius では，API として HTML 等のマークアップ文書と連携して音声インタフェースを適用するための規格である Speech Application Language Tags (SALT) へ対応した．

2.2.1 言語関連の機能拡張

音声対話システムでは，数字認識からディクテーションまで様々な認識タスクが要求される．このような多様なタスクに柔軟に対応するために，主に言語モデル関連で以下のような機能拡張がなされた．

- 任意長 N-gram のサポート
- 孤立単語認識のサポート
- 単語グラフ，confusion network 出力のサポート
- Julius/Julian の統合
- 複数文法・複数言語モデルの同時認識

特に，2007 年 12 月に公開されたバージョン 4.0 [2] からは，ソースの構成が大幅に改善され，言語制約がモジュール化された．N-gram，文法，および辞書のみを用いる孤立単語認識を，一つのエンジンで切り替えて使用できる．このため，従来の Julian は Julius に統合された．また，ユーザ定義の言語制約関数の埋め込みや，複数の言語モデルを同時に用いた音声認識を行えるようになった．一部の機能は SRM で未対応であるが，これらの改善によって，対話の流れに沿った言語モデルの切り替えや複数のタスクの同時認識など，より柔軟な音声対話システムの構築に寄与するものと期待される．

2.2.2 入力に対する頑健性の向上

雑音を含む実環境における音声認識システムの安定動作と精度向上を目標に，以下のような機能追加および改善を行った．

- 音声区間検出 (VAD) の強化
- MAP-CMN および実時間エネルギー項正規化
- バッファリング改善による音声入力の低遅延化

特に，VAD の強化では，まず Gaussian mixture model (GMM) に基づく発話単位での入力音識別と入力棄却 [3] が実装され，バージョン 4 でフレームベースの GMM に基づく音声区間検出，およびデコーダ内の単語仮説情報を用いて区間検出を行うデコーダベース VAD [4] が実装された．これらは不要音による誤動作を防ぎ，雑音環境下での動作をロバストにした．

2.2.3 性能・安定性の改善

ソフトウェアとして以下の改善が行われた．

- 高速化
- メモリ管理の改善
- Windows での安定動作
- ソースの統合，ドキュメンテーション

高速化では，MFCC 計算における sin, cos 演算のテーブル化，および音響尤度計算における全ての除算を乗算に変換した．特に後者の改善により，全体の認識処理時間を，標準の Phonetic tied-mixture (PTM) トライフォンで 20%，通常のトライフォンで，PC 上で 40% 程度高速化できた．また，木構造化辞書や N-gram の構造の最適化，ワークメモリの管理の改善により，よりコンパクトに安定して動作するようになった．また，認識システムのより深い理解の一助となるべく，Doxygen を用いたソースコードのドキュメンテーションも整備された．

2.3 音声合成モジュール: SSM

2.3.1 GalateaTalk

音声合成モジュール SSM は，日本語テキスト音声合成システムとして単体で動作する GalateaTalk として開発されている．入力コマンドの解析部，音声波形を合成する音声合成エンジン (波形生成部)，音声出力を音声出力部を実装した gtalk が，形態素解析を行う chasen および音韻交替処理やアクセント結合処理などを行う chaone を内部で呼び出すことによって，GalateaTalk は実現されている．

GalateaTalk の波形生成部では，HMM に基づいた音声合成 [5] を用いている．音声合成のための形態素解析では，発音形およびアクセントに関する素性の情報を得ることが必要であることから，UniDic プロジェクト [6] によって開発された辞書を利用している．

GalateaTalk では，対話音声の合成を行うことを指向して以下の機能が実現されている．

(1) 音声出力の途中での停止： 音声対話を行っている利用者からの割り込み発話があった場合に，合成音声の出力を停止すると同時に，停止時点までに出力された音素系列を知ることができる．
(2) 話者の変更： 1 発話内でも，部分的に話者を変更して発声することができる．また，周波数ワーピング係数を変更することによって，簡単な声質の変更も容易に行える．
(3) 韻律の柔軟な制御: 発話内容を記述する日本語テキストに，「日本語テキスト音声合成用記号の規格 (JEIDA-62-2000)」によって提案された「テキスト埋め込み制御タグ」[7, 8] を埋め込むことにより，韻律を部分的に変更することができる．
(4) 顔画像出力との同期： 発話における音素時間長のデータを顔画像生成のモジュールと共有することによって，画像における口唇の動きと合成音声とを同期させて生成する．

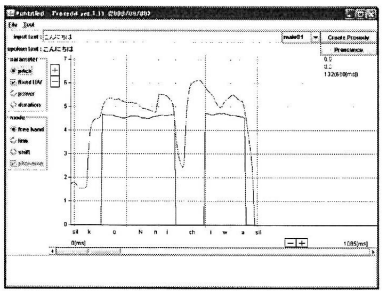

図 2: Prosedit の動作画面

2.3.2 カスタマイズツール

多様な対話音声合成の実現を目指して，話者モデルの構築ツール VoiceMaker と合成音声における韻律を手修正するツール Prosedit が提供されている．

VoiceMaker では，数十〜数百の文音声を指示に従って発声することによって，GalateaTalk で利用可能なその人の話者モデルを自動的に構築できる．

Prosedit は，GalateaTalk の合成した音声の韻律 (基本周波数，パワー，音素時間長) を GUI インタフェースを用いて，変更できるツールである．図 2 に動作画面を示す．

2.4 顔画像生成モジュール: FSM

2.4.1 基本機能

正面顔画像から 3 次元顔モデルを生成可能なエージェント生成ツール FaceMaker によって，任意のエージェントを作成可能である．顔画像合成モジュール FSM は，作成されたエージェント顔モデルの表情・発話などのアニメーションをリアルタイムに生成することができる．任意のエージェントモデルに対し，一般的な表情変化・発話を行わせるため，標準顔モデルを用意した．標準顔モデルはヒトの顔部位の構造を考慮した 3 角形ポリゴンで構成され，ポリゴンを構成する頂点群を移動させることで，さまざまな表情を生成可能である．表情のモデル化には，表情記述規則 FACS（Facial Action Coding System）[9] を導入し，44 個の基礎表情（AU: Action Unit）として定量化した．AU の移動量および，移動方向をパラメータ化することで，容易に複雑な表情を構築することが可能である．口形状のモデル化には，視覚素を日本語発話における発話の最小単位とし，視覚素 13 種の移動量，移動方向をパラメータ化した．図 3 に典型

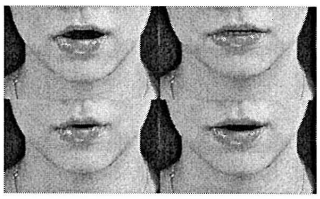

図 3: 生成された母音口形状の例

的な母音の口形状を示す.

2.4.2 機能拡張

ISTC における FaceMaker，FSM の主な機能拡張は以下のとおりである.

(1) 英語発話対応： 視覚素を 13 種類から 27 種類へ拡張し，英語の発話に必要な口形状を用意した．視覚素の追加により，英語文章の発話に必要な英語音素と音素継続長のコマンドを FSM へ送ることで，日本語発話と同様なリップシンクアニメーションを生成することが可能となった．また英語・日本語モードの切り替えは，設定ファイルを書き換えることで実現した.

(2) リップシンク精度向上： 音声合成モジュールから出力される合成音と発話アニメーションのリップシンク精度を向上させるため，音声合成が管理するローカル時間と，FSM が管理するローカル時間の補正が発話中に行えるように改良した．また発話開始時間のオフセット時間を設定できるコマンドを追加した.

(3) 出力画面キャプチャ機能： 発話アニメーションの出力結果を連番の画像ファイルとして保存できる機能を追加した．キャプチャ時のフレームレートは任意に設定可能である.

(4) ユーザインタフェース改良 (FaceMaker)： Galatea プロジェクトで開発したエージェント生成ツールは操作方法が特殊であるため，一部ユーザから操作が難しいとの意見があった．今回 UI を改良し，ユーザビリティの向上に努めた．改良に伴い，OpenGL 用 GUI ツールキットとして Qt を採用した.

その他，バグフィックス・性能向上のためのソースコード見直しを実施した．今後の開発方針の一つとして，FaceMaker のエージェント生成時間の短縮が挙げられる．現在標準顔モデルと正面顔画像はマニュアルで整合を行うため，作成に時間を要する．多くのエージェント顔モデルを短時間で生成するため，RBF

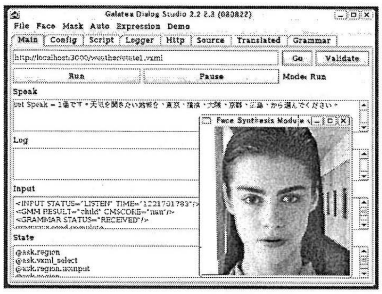

図 4: Galatea Dialog Studio の実行画面

（Radial Basis Function）を用いた新たな整合手法を導入予定である.

2.5 対話制御モジュール: AM

2.5.1 Linux 版

Linux 版統合システムにおいては，Perl 言語で記述された AgentManager がサブモジュール間の通信を行い，対話処理系が各サブモジュールを制御する．対話処理系 (Galatea Dialog Studio) は音声対話記述言語の標準規格である VoiceXML に基づいて開発されており Java 言語で実装されている．図 4 に動作画面の例を示す．2003 年の IPA 最終版リリース時には AgentManager を呼び出して制御する最低限の機能のみを備えていた [10]．その後，現在までに各サブモジュールの改良に対応しつつ以下のような改良が行われた.

(1) VoiceXML コンテンツの作成を容易にするための音声認識の文法記述の改良

(2) エージェントの動作や表情をより自然に制御したり，発話と表情変化を並行して行うための機能拡張

(3) オーディオファイルの出力など標準準拠の拡張

(4) GUI による操作支援，システム状態表示，ログやエラーなどの表示機能の強化

(5) 実装のリファクタリングと将来のプラグイン対応への準備，一部機能の Windows 対応

(6) PHP および Ruby on Rails などウェブアプリケーション開発技術への対応

(7) Ubuntu Linux および Knoppix など新しい Linux 実行環境への対応

特に Ruby on Rails は VoiceXML コンテンツの作成の支援に大きく貢献し，高度なマルチモーダル対話システムの設計にも多くの示唆を与えるウェブ

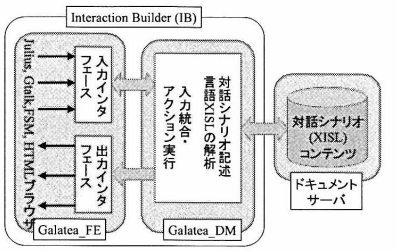

図 5: Windows 版対話制御モジュールの構成

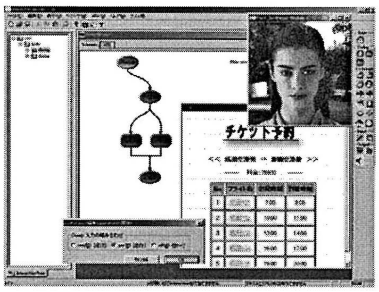

図 6: Interaction Builder の実行画面

アプリケーション開発ツールである．モダリティ非依存の記述は Ruby on Rails の機能をそのまま使用し，モダリティに固有の部分のみを HTML ではなく VoiceXML で記述する，といったことが可能になる．今後は Ruby on Rails による VoiceXML 開発の支援についてさらに検討を進める．

2.5.2 Windows 版

Windows 版の対話制御モジュールは，各種エンジンをコントロールする Galatea_FE，対話制御モジュールである Galatea_DM，および対話シナリオ作成プロトタイピングツールである Interaction Builder(IB)[11] から構成される．全体のシステム構成を図 5 に示す．

Galatea_FE は音声認識エンジン Julius，音声合成エンジン Gtalk，顔画像合成エンジン FSM を起動し，各モジュールとソケット通信によって命令を授受する．また，web ページを表示する機能を備えており，ポインティングと音声を用いたマルチモーダル入力を受付可能にしている．

Galatea_DM は XISL[12] という言語で記述された対話シナリオに従って対話を進行させる．XISL は XML ベースのマルチモーダル対話記述言語で，状態遷移，マルチモーダル入出力，算術演算や条件分岐を記述することができる．Galatea_DM では XISL の対話進行に関する部分を解析・実行し，入出力に関する部分は Galatea_FE にソケット通信で伝達することにより，入出力と対話進行の分割管理を実現している．

IB は GUI 操作によって XISL 文書を自動生成するためのツールである．ユーザは対話部品のアイコンをドラッグ&ドロップするだけで対話の流れを容易に構築することができる．また，典型的な対話の流れについてはウィザード機能を用いて簡単に対話シナリオを構築することができる．図 6 に IB の実行画面を示す．

3 音声対話のための MMI 記述言語標準化

ISTC のマルチモーダル対話 (MMI) 記述言語策定ワーキンググループでは，MMI 記述言語の標準化を目指して，まずユースケースの取り纏めを行い，その後，要求仕様の抽出，およびシステムアーキテクチャの提案を行った [13].

3.1 MMI システムのユースケース

まず対象とするアプリケーションを具体化するために，各企業／大学においてこれまで構築してきた MMI システムを念頭に，以下の 8 種類のユースケースを作成した．

- オンラインショッピング
- 音声によるディレクトリ検索
- サイト検索
- ロボットとの対話
- 対話エージェントとの交渉
- 音声情報案内システム
- エリアガイド
- カーナビ目的地設定

各ユースケースには対話シナリオ，利用モダリティ，システムに必要な機能等が記載されている．詳細は [14] の web サイトを参照されたい．

3.2 MMI システムの要求仕様

上述のユースケースに基づいて，マルチモーダル対話システムの要求仕様を抽出してまとめた．以下に一覧を示す．

(1) 一般的な要求
(2) 入力モダリティに関する要求
(3) 出力モダリティに関する要求
(4) アーキテクチャ，統合，同期について

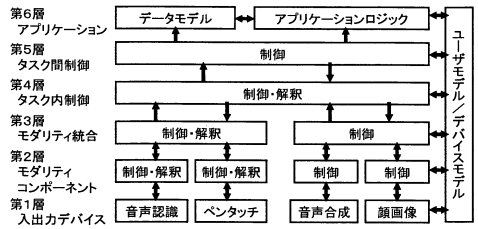

図 7: MMI システムの 6 階層モデル

(5) 実行時と配置
(6) 対話制御について
(7) フォーム・フィールドのハンドリング
(8) アプリケーション・外部モジュールとの連携
(9) ユーザ情報，環境情報
(10) 開発者の視点から見た機能
(11) アプリケーションとセッション
(12) ECMAScript の利用

上記の (1)〜(5) は W3C においてまとめられた MMI システムの要求仕様にも含まれる項目であるが，(6)〜(12) はユースケースに基づいて ISTC が独自に組み込んだものである．要求仕様は [15] の Web サイトにおいて公開している．

3.3 MMI システムアーキテクチャ

MMI システムでは各種モダリティの制御や認識結果の統合，対話制御など，多段階の処理が必要であることから，多階層モデルが検討されることがある．従来研究ではモダリティ制御と対話制御からなる 2 階層のモデル（例えば W3C のモデル）が検討されることが多かったが，ISTC では図 7 に示す 6 階層からなるアーキテクチャを検討した．

このモデルでは，開発者は必ずしも各階層毎にモジュールを実装する必要は無い．例えば 2 層と 3 層，あるいは 3 層〜5 層を一つのモジュールとして開発することも容認される．本モデルでは，従来と比べて細かく MMI システムの機能を分割しているため，多様な切り分けによるモジュール実装が可能である．

4 まとめ

ISTC におけるこれまでの成果として，擬人化音声対話ツールキット Galatea とマルチモーダル対話 (MMI) 記述の標準化について述べた．

Galatea に関しては，国際化の方針決定と英語等への対応，Live CD/DVD 版の Galatea の作成・公開，ドキュメントやライセンス記述の整備，などが今後の課題として挙げられる。また，MMI システムのアーキテクチャについては，情報処理学会の試行標準としての公開を目指して取りまとめ作業を進めており，W3C の MMIWG とも連携を取りつつ，国際標準への盛り込みも検討している．ISTC の活動は，2009 年 3 月で終了する予定であり，その後も多くの開発者に関心を持っていただくために，これらは重要な課題である．

謝辞

本プロジェクトでソフトウェア開発にご協力いただいている音声対話技術コンソーシアム (ISTC) の実行委員，ならびにコンソーシアム会員に感謝する．

参考文献

[1] http://www.astem.or.jp/istc/
[2] 李 晃伸, “大語彙連続音声認識エンジン Julius ver.4”, 信学技報, SP2007-54, **107**, 406, pp.307–312 (2007).
[3] A. Lee, et al., “Noise robust real world spoken dialogue system using GMM based rejection of unintended inputs,” ICSLP2004, pp.847–850 (2004).
[4] 酒井, 他, “実環境ハンズフリー音声認識のための音響モデルと言語モデルに基づく音声区間検出と認識アルゴリズム”, 信学技報, SP2007-17, **107**, 116, pp.55–60 (2007).
[5] 益子, 他, “動的特徴を用いた HMM に基づく音声合成”, 信学論 (D-II), **J79-D-II**, 12, pp.2184–2190 (1996).
[6] http://www.tokuteicorpus.jp/dist/
[7] 蓑輪, 他, “JEIDA 日本語テキスト音声合成用記号”, 日本音響学会秋季講演論文集, 2-1-5, pp.183-184 (2000).
[8] (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).
[9] P. Ekman, W. V. Friesen, “Manual for the Facial Action Coding System and Action Unit Photographs”, Palo Alto, CA: Consulting Psychological Press (1978).
[10] 西本, 他, “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).
[11] K. Katsurada, et al., “Interaction Builder: A Rapid Prototyping Tool for Developing Web-Based MMI Applications”, IEICE Trans. Inf. & Syst., **E88-D**, 11, pp.2461–2468 (2005).
[12] 桂田, 他, “MMI 記述言語 XISL の提案”, 情報処理学会論文誌, **44**, 11, pp.2681–2689 (2003).
[13] 新田, 他, “マルチモーダル対話システムのための階層的アーキテクチャの提案”, 情報処理学会研究報告 2007-SLP-68, pp.7–12 (2007).
[14] http://www.astem.or.jp/istc/ISTC-SIG-MMI/index.html
[15] http://www.astem.or.jp/istc/ISTC-SIG-MMI/meeting12/MMI-Requirement3.pdf