ヒューマンインタフェース　97－13
音声言語情報処理　40－13
（２００２．２．１）

対話システムにおけるタスク記述とプロトタイプ作成支援

西本 卓也†　新田恒雄‡　足立裕秋‡　桂田浩一‡

京都工芸繊維大学 工芸学部†　豊橋技術科学大学 大学院工学研究科‡

〒 606-8585　京都市左京区松ヶ崎御所海道町　E-mail : nishi@dj.kit.ac.jp †
〒 441-8580　愛知県豊橋市天伯町雲雀ヶ丘 1-1　E-mail : nitta@tutkie.tut.ac.jp ‡

あらまし　本報告では，情報処理振興事業協会 (IPA) の擬人化音声対話エージェント基本ソフトウエアにおける統合部のうち，タスク記述およびプロトタイプ作成支援について述べる。本報告では，擬人化音声対話エージェントを用いたマルチモーダル対話の記述方式として，VoiceXML をマルチモーダルに拡張する方法と，マルチモーダル対話用に設計された XISL を用いる方法の二つを紹介する。続いてタスク記述方式では，VoiceXML を中心に他の記述言語も実装容易なインタプリタの設計方法について述べる。これにより，タスクに適した対話記述を選択することが可能になる。最後に，マルチモーダル対話タスクとしてオンラインショッピングを例に取り上げ，XISL に基づくプロトタイプ作成支援システム (Interaction Builder (IB)) を説明する。

キーワード　マルチモーダル対話，音声対話，タスク記述，VoiceXML，XISL，プロトタイピング

Task Description and a Prototyping Tool for Interactive Systems

Takuya NISHIMOTO †, Tsuneo NITTA ‡, Hiroaki ADACHI ‡, and Kouichi KATSURADA ‡

Department of Electronics and Information Science, Kyoto Institute of Technology †
Graduate School of Engineering, Toyohashi University of Technology ‡

Matsugasaki, Sakyo, Kyoto, 606-8585 JAPAN　E-mail : nishi@dj.kit.ac.jp †
1-1 Hibariga-oka, Tempaku, Toyohashi, 441-8580 JAPAN　E-mail : nitta@tutkie.tut.ac.jp ‡

Abstract　This paper describes a method of task description and a prototyping support system for the IPA (Information Processing Association)-supported fundamental software of an anthropomorphic agent with voice interaction facilities. We propose two approaches for describing multimodal interaction, namely, VoiceXML extended to multimodal interaction and XISL designed for multimodal interaction. In this report, firstly, a task description language and its interpreter are introduced in which not only VoiceXML but also other description languages are easily implemented. Secondly we describe a domain-specific prototyping support system named Interaction Builder (IB) based on XISL for on-line shopping systems.

Key words　Multi-Modal Interaction, Spoken Dialogue, Task Description, VoiceXML, XISL, Prototyping

1　はじめに

本報告では，情報処理振興協会 (IPA) の擬人化音声対話エージェント基本ソフトウエア開発（以下，IPA プロジェクトと呼ぶ）[1] における統合部 [2] のうち，タスク記述とプロトタイプ作成支援の 2 つを報告する。統合部の中で，音声認識・合成，および顔合成モジュールを統合するエージェントマネジャについては，他の報告 [3] を参照されたい。

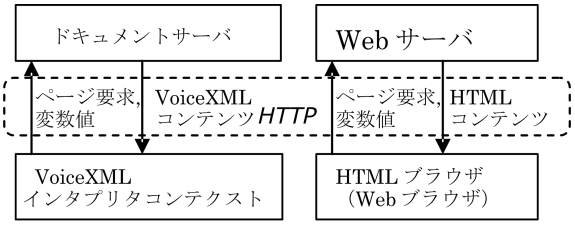

図 1: VoiceXML と Web のアーキテクチャの比較

2　対話タスク記述と対話インタプリタ

人間性豊かな擬人化音声対話エージェントを構築する場合，自然で多様な表情が可能な顔表情出力，感情を表現できる音声合成といった要素以外に，意味理解や推論などの自然言語処理能力なども重要になる。しかし，幅広い応用を前提とする汎用ツールキットでは，対話タスクにおける自然言語処理は必ずしも必要でなく，またアプリケーションにゆだねられる部分が大きい。

本報告では擬人化音声対話エージェントの制御を，入力と出力の対応関係として記述する対話タスクとして表現し，それを実行するものとして扱う。さらに，擬人化音声対話エージェントと人間の対話をマルチモーダル対話 (Multi-Modal Interaction, MMI) と捉える。

MMI の実現については，その動作環境や用途の違いからいくつかの方式が考えられる。ここでは，音声対話記述言語 VoiceXML とそのマルチモーダル拡張について述べ，他のマルチモーダル対話記述方式を含めて検討を行う。また，これらの対話記述言語に基づいて動作する対話インタプリタの設計について述べる。

2.1　VoiceXML のマルチモーダル拡張

IPA プロジェクトでは音声対話記述言語 VoiceXML を対話タスク記述言語の 1 つとして検討している。特に，電話環境に固有の機能を削除しつつ，擬人化音声対話エージェント独自の機能を活かす機能拡張を行っている。

VoiceXML は主に電話を利用した音声応答サービスの提供を支援するマークアップ言語であり，XML をベースとした言語仕様になっている。AT&T, IBM, Lucent, Motorola によって設立された VoiceXML Forum が仕様を作成しており，Version 1.0 が 2000 年 3 月に公開された。現在は World Wide Web Consortium (W3C) における音声対話記述言語として標準化が行われている[1]。

VoiceXML のアーキテクチャでは，データベース検索や推論などの処理は VoiceXML インタプリタの外部にあるドキュメントサーバに依頼する。これは Web で動的なページを生成する処理系に似ており，VoiceXML インタプリタは Web と同じ HTTP プロトコルでドキュメントサーバと通信する（図 1)。

このような構成を取ることにより，VoiceXML インタプリタはユーザ・インタフェースに関する処理に専念し，アプリケーション固有の処理には Web サーバと共通の手法（CGI, PHP, Servlet など）を利用できる。

現在のバージョンの VoiceXML は，視覚的な手段を持たない電話音声応答システムでの利用を前提としており，入力として仕様に含まれているのは音声または DTMF（数字キー入力）である。出力は音声合成または録音音声（オーディオファイル）の再生が可能である。

対話はフォームおよびメニューを単位とする状態遷移によって行われる。入力はフォームに含まれる各フィールドの入力またはメニューの選択肢からの項目選択が基本単位となる。ABNF 形式および XML 形式のグラマー（音声認識文法）に，意味情報を表すタグを付与することができる。タグを用いることで 1 発話で複数のフィールドへの入力を行うことが容易になる[2] 。

VoiceXML ではスロットフィリング形式の対話記述が容易に記述できるが，タグ付きグラマーの利用や，実行中のフォーム以外の対話に移動するリンク機能などによって，混合主導的な音声対話も実現で

[1] http://www.w3.org/Voice/
[2] http://www.w3.org/TR/2001/WD-speech-grammar-20010820/

```
<menu>
  <prompt>
   ノートパソコンには
   <play act="point">
     αタイプ，βタイプ
   </play>
   がございます。
   <emotion type="happy">
     どちらにしますか？
   </emotion>
  </prompt>
  <choice next="xx" img="a.gif">
     αタイプ
  </choice>
  <choice next="xx" img="b.gif">
     βタイプ
  </choice>
</menu>
```

図 2: VoiceXML 拡張によるエージェント制御の例

きる。

VoiceXML のモデルに沿ったマルチモーダル対話であれば，メニューやプロンプトなどの概念を拡張することで実現できる [4]。この際，なるべく低レベルの実装に依存した処理を隠蔽し，上位概念での記述ができることが好ましい。図 2 にエージェントが音声合成を行うプロンプト記述例を示す[3]。選択肢の文字や画像を指差すジェスチャを行ったり，テキストの特定の部分のみ感情を付与することを想定している。

2.2 MMI 記述方式の比較

VoiceXML を用いた対話システム構築は，特にシステム主導対話記述の可読性が高く，ドキュメントサーバの実装も容易である。しかし，GUI による操作と連携した音声対話を行ったり，ジェスチャ認識と統合して対話を制御する，といった処理を行うには，VoiceXML のモデルだけでは不十分である。

VoiceXML では対話の状態遷移に同期したユーザ入力処理を前提としている。従って，対話コンテクストと無関係に生成されるイベント（例えば，誰かがエージェントの前に立ったことをカメラが検出した，など）を処理するためには，VoiceXML に対する上位レベルのコンテクスト制御言語が必要となる。

また，VoiceXML による開発においては，コンテンツとプレゼンテーションの分離はドキュメントサーバにおいて行われることが望ましい。しかし，現実には録音音声や音楽を用いるなどの理由により，コンテンツとプレゼンテーションの分離が困難になりがちである。

コンテンツとプレゼンテーションを分離して汎用性の高いマルチモーダル対話システムを効率的に構築することは，今後重要になると考えられる。そこで IPA プロジェクトでは MMI 記述言語として新たに XISL を提案している。XISL を用いた開発については第 3 章にて述べる。

Web で普及しているコンテンツ記述言語を拡張する提案もある。HTML を拡張して MMI を実現する規格として，Microsoft などにより SALT(Speech Application Language Tags) が提案されている[4]。ストリーミングコンテンツ記述言語 SMIL [5] のマルチモーダル拡張も提案されている [5] 。

マルチモーダル・プレゼンテーションの記述言語も，マルチモーダル対話の出力として利用できる。キャラクタエージェントを用いたプレゼンテーション記述言語としては MPML（Multimodal Presentation Markup Language）が提案されている [6] [6] 。また，テレビ番組を記述可能な言語として TVML が提案されている[7] 。

2.3 対話インタプリタの設計

現在 IPA プロジェクトにおいて開発中のマルチモーダル対話インタプリタ Phoenix（仮称）は，擬人化音声対話エージェントがさまざまなアプリケーションや研究で用いられることを想定して，VoiceXML に基づく対話タスク記述言語だけでなく，ニーズに応じた対話タスク記述言語の拡張も考慮している。例えば新たなモダリティを追加する場合などにおいて，新しい言語を容易に実装できること，既存の言語の拡張が容易であること，などが望ましいからである。

[3]後述する MPML を参考にしている。

[4]http://www.saltforum.org/
[5]http://www.w3.org/AudioVideo/
[6]http://www.miv.t.u-tokyo.ac.jp/MPML/jp/
[7]http://www.strl.nhk.or.jp/TVML/

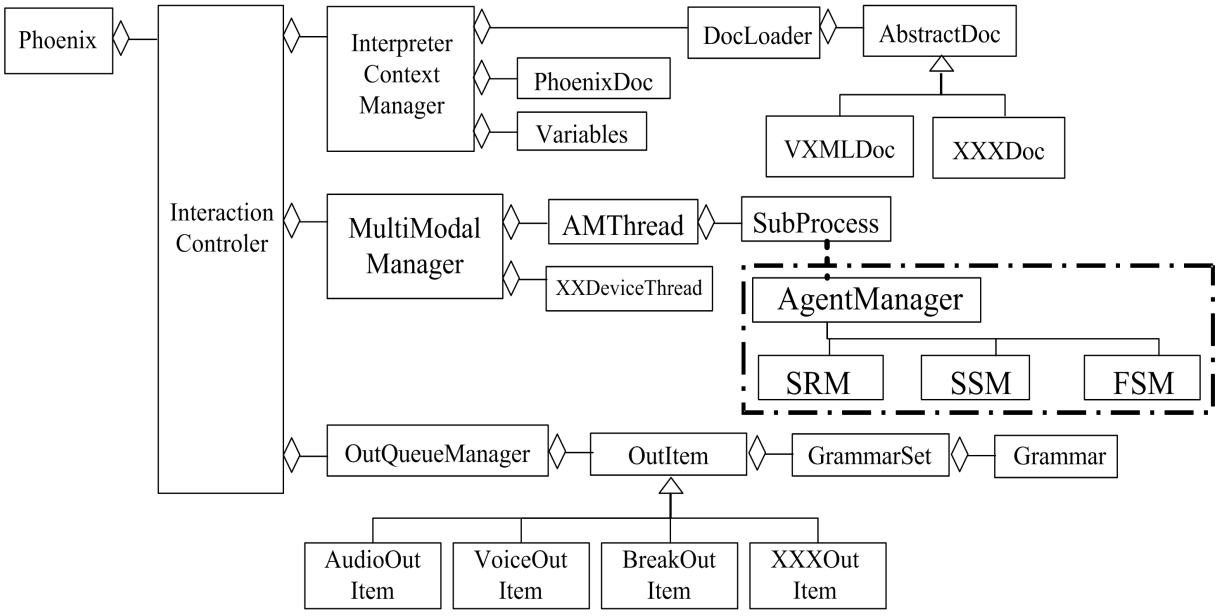

図 3: マルチモーダル対話インタプリタの主要クラス構成

Phoenix は Linux 環境で動作し，エージェントマネジャを外部プロセスとして呼び出す。実装には Java 言語を用いている。図 3 に，現在設計中のクラス構成を示す。主要なクラスとその機能は次の通りである。

マルチモーダル制御部 (MultiModalManager)：エージェントマネジャを通じて音声認識，音声合成，顔画像の各サブモジュールの制御を行うラッパークラスである。エージェントマネジャは独立したスレッドとして動作し，サブモジュールが非同期的に生成する音声認識結果などのイベントを他のクラスに受け渡す。将来的には GUI など他の入出力手段に対応するスレッドも管理し，他のプラットフォームでの動作や，他のエンジンに対応した対話制御が容易になるような抽象化を目指している。

タスク記述言語処理部 (DocLoader)：VoiceXML などのファイルをマルチモーダル制御部の処理に対応した中間コードに変換する。実際のマルチモーダル対話は，出力キューの状態変化に応じて中間コードを逐次処理する状態遷移モデルで実行される。例えば VoiceXML におけるフィールドの実行は，以下のような中間コードに変換することができる。

1. プロンプトとグラマーをキューに追加
2. 入力がなかった場合に実行するイベントの登録
3. キューを実行（プロンプト再生および入力待ちを並行して行う）

また，グラマーの解釈および登録を行う他，コンテンツ開発者に対してタグの文法や用法の誤りなどを報告する機能を持つ。新たなタスク記述言語への対応が，タスク記述言語処理部の拡張だけで可能になるような設計を目指している。

対話コンテクスト管理部 (InterpreterContext Manager)：中間コードの実行状態を保持し，変数などの管理を行う。また，新たなドキュメントへの移動に伴ってネットワークとの非同期通信処理やタスク記述言語処理部の呼び出しを行う。

出力キュー管理部 (OutQueueManager)：対話コンテクスト管理部からの要求によって新たな出力キュー項目（音声合成の出力，話者や表情の切り替え，オーディオファイルの出力，一定時間の休止，など）の追加を行い，その情報を管理する。マルチモーダル制御部からの出力完了イベントによって，キューから次の項目を取り出して出力を行う。なお，VoiceXML コンテンツの実行においては，出力の状態変化に応じて受理できるグラマーが変化すると考えられる。そこで，出力キューの各項目は対応するグラマーの状態を保持する設計になっている。

3　プロトタイピングツール

ここでは，オンラインショッピングを対象とした domain-specific prototyping tool について説明する。このツール（以下，Interaction Builder(IB) と呼ぶ）は，IPA プロジェクトで開発中のモジュール（音声認識・合成，顔合成）を含む，マルチモーダル対話 (MMI) 環境を短期間に構築することを目的としている。現在は，多様な MMI を記述するための言語，XISL(Extensible Interaction-Sheet Language) [7] を独自に策定し，この仕様に沿って MMI システムのプロトタイピングを進めている。

3.1　MMI 記述言語 XISL

XISL は XML コンテンツに対する MMI を記述する言語である。XISL では，複数の dialog から対話シナリオを構成し，各 dialog は対話の最小単位を表す exchange から構成される。また，各 exchange は operation(入力記述部) と action(アクション記述部) をそれぞれ一つずつ持つ。単一の入出力記述は，input と output により表される。Input にはシステムが受け付けるユーザからの入力を，output には対応する動作を記述する。さらに operation や action 内には，複合入出力を制御するタグ (sequential：逐次，parallel：同時並行，alternative：択一) を用意している。

XISL 仕様に基づく応用システム開発では，XML（コンテンツ），XSL（ビュー），XISL（インタラクション）という三つの言語記述でドキュメントを制作することになる。このため開発者が GUI 操作によって，システムを構築する環境が必要になる。

3.2　オンラインショッピングシステムの概要

今回は PC 端末上で動作するオンラインショッピングシステム (以下 OLS と呼ぶ) をプロトタイピングの対象とした。このシステムは，入力モダリティとして音声とタッチパネル（ポインティング，移動など）を，出力モダリティとして音声 (TTS)，サウンド，ディスプレィ（ブラウザ；動画含む），および擬人化エージェント（動作，TTS，バルーン）を持つ。なお音声入出力は，現在，市販ソフトを，またエージェントは MS エージェントを利用しており，順次 IPA 開発のモジュールに置換える予定である。

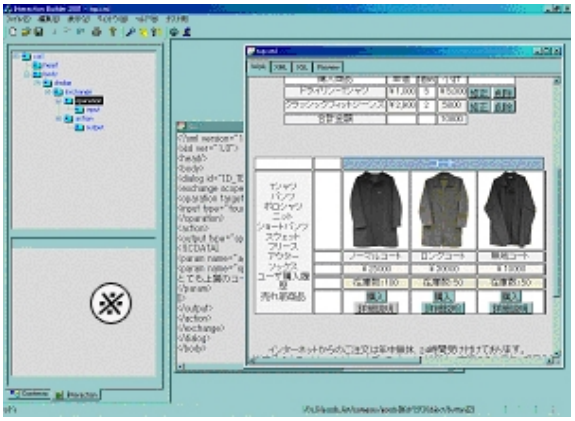

図 4: Interaction Builder の実行画面

OLS システムでは，商品と顧客データ (XML)，スタイル (XSL)，および対話シナリオ (XISL) の 3 つの構成要素を独立に扱っている。これにより保守性と再利用性の高いシステムを実現することができる。さらに，システム開発時には，新たなモダリティの追加のし易さ，対話シナリオの詳細な記述が可能といった，XISL 言語仕様から来るメリットを活用できる。しかし，反面，データ構造の把握，記述量の増加といったドキュメント製作者負担が少なくない。また，端末に依存したモダリティの動作記述習得も必要である。

3.3　Interaction Builder (IB)

システム開発者の負担を軽減するために，開発支援ツール IB を開発している（図 4 参照)。IB は，ブラウザの 1 画面（ページと呼ぶ）毎にインタラクションを記述する。以下では IB を用いた MMI 開発手順を説明する。

step-1: 図 5 の Page Window (PW；XML と XSL から生成される画面) を開く。ブラウザ画面は PW 上部のタブ (XML, XSL) を指定し，各々テキストエディタを使用して作成する。

step-2: PW から Work を指定し，ページに対するインタラクションを貼り付ける作業画面 (Work View) に切替える。同時に Interaction Window (IW) が立ち上がる（図 4 参照)。

step-3: IW には対話記述結果が表示され，テキス

図 5: Page Window

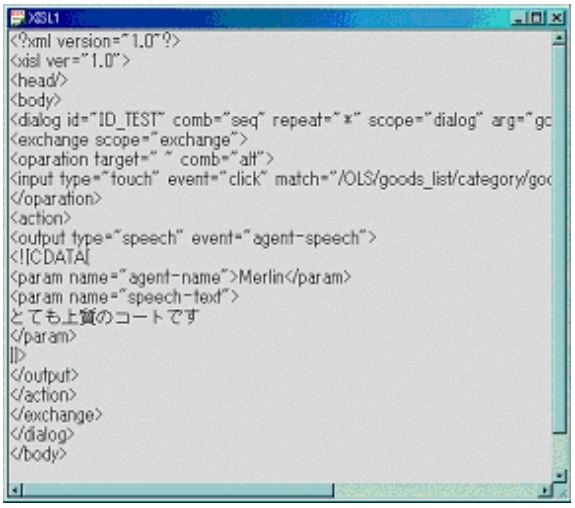

図 6: Interaction Window

トエディタで修正することができる（図 6 参照）。対話記述作業は，まず図 4 左上の子ウィンドウにページ内対話のスケルトンが表示され，ダイアログボックスを開きながら詳細を作成する。結果は IW に反映される。

1 ターン（Exchange）内のユーザ操作 (operation) とシステム動作 (action) は，Work View 上で対象オブジェクトを選択した後，図 4 上部の tool bar にある入出力モダリティ指定 box (Input Tool Box, Output Tool Box) を開いて選択・記述（音声入力では語彙・文法など，TTS では文・合成仕様など）を行う。なお，音声対話などでオブジェクトを指定しない場合は，デフォルトでページ自身がオブジェクトとなる。

対話シナリオの構成や動き（遷移）を観る手段についても，幾つか用意している。図 4 左下にはモダリティの利用形態（Seq./ Par./ Alt）が時間軸と共に示されている。このほかページ間の遷移や，ページ内の遷移をグラフで表示する機能などを提供する予定である。

4 おわりに

IPA 擬人化音声対話エージェント基本ソフトウエア開発における統合部のうち，対話記述とプロトタイピングツールの二つについて現状を報告した。今後は対話記述機能を拡張すると共に，開発予定の各モジュールを組込み，統合制御ソフトウエアとして完成させる予定である。また，インタプリタおよびプロトタイピングツールについても順次提供していきたい。

参考文献

[1] 嵯峨山茂樹他: ”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] 新田恒雄他: ”対話システムにおけるモジュール統合とプロトタイピング,” 情報処理学会研究報告 2000-SLP-33-5, Oct. (2000).

[3] 川本真一他: ”擬人化音声対話エージェントツールキットの基本設計,” 情報処理学会研究報告 2001-SLP-40-11, Feb. (2002)

[4] 植田喜代志他: ”VoiceXML のマルチモーダル化の検討”, 情報処理学会研究報告, 2001-SLP-38-7, pp.43-48, (2001).

[5] Beckham, et. el: Toward SMIL as a Foundation for Multimodal, Multimedia Application, Eurospeech 2001; pp.1363-1367, (2001).

[6] 筒井貴之他：キャラクタエージェント制御機能を有するマルチモーダル・プレゼンテーション記述言語 MPML, 情報処理学会論文誌, Vol.41, No.4, pp.1124-1133 (2000.4)

[7] 小林聡他：”マルチモーダル対話記述言語 XISL の提案”, 情報処理学会研究報告, 2001-SLP-37, pp.43-48, (2001).