

フレーム駆動によるマルチモーダル対話制御方式

荒木 雅弘[†] 秋田 祥史[†]

京都工芸繊維大学 工芸学部 電子情報工学科[†]

概要

本稿ではフレーム駆動による対話制御方式を用いたマルチモーダル対話の記述手法を提案する。本手法では、フレーム単位でマルチモーダル情報のコントロールを行い、主モード(音声)および副モード(HTML によってコントロールされるモード)のインタラクションをフレームおよびスロット毎のプロセスとして記述することによって、可読性・タスク移植性の高いマルチモーダル対話制御方式を実現した。

1. はじめに

近年の音声認識・合成技術の進歩に伴い、音声入出力に、ポインティングデバイス・キーボードなどによる入力や、文字・静止画・動画などの出力を組み合わせたマルチモーダル対話システムの開発が盛んに行われている。従来は研究室レベルの独自実装方式であったが、近年では実際の普及を目前にして、それぞれの目的に応じた記述言語が提案されている。

マイクロソフト社の SALT[1]は、HTML ページに音声の入出力等を記述したタグを埋め込み、スクリプト言語でインタラクションを記述するものである。主に HTML におけるフォームへの入力に音声を補助的に用いるような応用が適している。

XHTML+Voice[2]は、XHTML ページで発生するイベントのハンドラとして VoiceXML が記述可能であるので、SALT と同様の目的で用いることができる。

豊橋技術科学大学で開発されている XISL[3]はコンテンツ・ビュー・インタラクションの分離を目指して設計されている。

これらのマルチモーダル対話記述言語は、中心となるモードのインタラクション記述に、他のモードのコントロールとタスクレベルの対話制御情報が混在した形式を取っていることが多いので、可読性・保守性の面で問題がある。

そこで我々は、フレーム駆動による対話制御方式を用いたマルチモーダル対話の記述方式を提案する。本手法では、フレーム単位でマルチモーダル情報のコントロールを行い、主モード(音声)および副モード(HTML によってコントロールされるモード)のインタラクションをフレームおよびスロット毎のプロセスとして記述することによって、可

読性・タスク移植性の高いマルチモーダル対話制御方式を実現した。

提案方式の利点は、(1)タスク知識をフレーム形式で表現するだけで対話制御が実現できること、(2)音声モードに VoiceXML、それ以外のモードのコントロールに HTML を用いており、それらを拡張することなくマルチモーダル対話が実現できることである。

以下、2 章ではフレーム駆動対話制御手法の概要を述べ、3 章ではこのフレーム駆動方式のマルチモーダル対話への拡張法を説明する。4 章では本手法で実現した対話システムの例を示し、5 章でまとめと今後の課題を述べる。

2. フレーム駆動による対話制御

フレームによって知識を表現し、その知識構造に基づいて対話を制御する手法は、Bobrow らによって提案され[4]、キーボードによる対話システム GUS が実現された。その後、このフレーム駆動対話制御方式を用いていくつかの音声対話システムが実現されたが[5]、[6]、適用されたタスクはひとつのフレームで対話が完結するような単純なものであった。

我々は、このフレーム駆動対話制御方式を拡張し、フレーム構造を木構造と見なして動的に展開する手法で、複数のゴールを持った話題や、ユーザが主導権を取ることによる話題の遷移に対応できる方式を提案した[7]。

対話で用いるフレームは話題フレームと呼び、図 1 に示す構造を持つ。以下に構成要素の機能を示す。

- FrameName: 話題フレームが表現する話題に対応する。
- FrameID: 動的に生成する話題木の節点として話題フレームが用いられる場合の節点番号
- P_frameID: 話題木中の親節点の節点番号
- State: 話題木の中でこの話題が終了したか否かを表現する。
- Slot: 話題フレームの表す話題の下位話題に対応する。

FrameName	State	[FrameID]		[P_frameID]
Slot1	[Value]	Before Process1	After Process1	[Semantic Markers]
Slot2	[Value]	Before Process2	After Process2	[Semantic Markers]
Slot3 daemon	[Value]	Before Process3	After Process3	[Semantic Markers]
Slot4	[Value :default]	Before Process4	After Process4	[Semantic Markers]

図 1 話題フレームの構造

また、スロットは任意の数だけ書くことができ、各スロットは以下の項目からなっている。

- Slot: 上述した下位話題に対応する。
- Value: 通常空欄で、対話により求められる。必要ならデフォルト値を書く。
- Before Process: スロット値を求めるための処理で、ユーザへの質問、データベース検索、他の話題フレームの接続など。
- After Process: スロット値を求めた後の処理で、次の優先順位のスロットの処理に移る、など。
- Semantic Markers: スロット値として採用できる概念（数値、場所、時間など）

これらのスロット間には処理の優先順位が付いており、明示的に遷移されない場合は、この優先順位に基づいて対話が進められる。

3. フレーム駆動方式のマルチモーダル対話への拡張

3.1 フレームとマルチモーダル情報の対応

2章で説明したフレーム駆動方式は、各フレームが話題の単位を構成しており、フレームの遷移が話題の遷移に対応する。また、スロットに値を埋める行為は、基本的には1回のやりとり（ダイアログボックスに対するマウスクリックや、音声プロンプトに対する音声入力）に対応するとみなすことができる（一発話で複数のスロット値を埋めることも可能である）。

一方、ユーザに地図などの静止画や製品の動画を見せながら、音声入力やマウス入力によって次の状態に移ってゆく典型的なマルチモーダル対話システムを考えると、ユーザに提示している静止画・動画などの情報は、時には複数のやりとり単位に渡って提示すべきものもある。

そこで、特定の静止画・動画などを提示している間を話題の単位とみなしてフレームで表現し、その話題単位を構成するやりとりを、スロットに

値を埋める行為であるとみなせば、フレーム（すなわち話題）の移動と静止画・動画の出現・消滅を同期させることによって、複数のやりとりの間は特定の情報を提示し、話題が移動すれば消滅するようなマルチモーダル情報の制御が行える。

一方では、日付入力のやりとりにおけるカレンダーの提示のように、やりとり単位で生成・消滅すべき情報も存在する。これらの情報はスロット単位に付属させることによって、やりとりの間だけ提示することができる。

3.2 マルチモーダル対話記述方式の検討

3.1節で述べたようなマルチモーダル対話制御機構を利用して、実際にマルチモーダル対話を記述する方法にはいくつか考えられる。

最も柔軟な方法は、全てをプログラミング言語で記述し、メディアの入出力はローレベルのAPI(Application Program Interface)を用いる方法である。しかし、この方法は可読性・保守性に劣る。

最も抽象度を高めた方法は、マルチモーダルインタラクションが直接記述可能なフレーム記述言語を定義する方法であるが、新規言語を考案することになるので、技術的なブレークスルーとみなされない限り普及は難しい。

そこで我々は、いくつかの既存言語を寄せ集め、できるだけオリジナルの言語仕様に変更を加えない形でマルチモーダル対話を実現することを試みる。

まず、フレームの表現言語としては、Semantic Web のオントロジー記述言語である OWL[8]を利用した。OWL はクラス-サブクラスの関係や、プロパティに関する制約などを記述可能で、フレームを記述する言語として適している。

また、本手法はフレームおよびスロットで対話の進行を管理するモデルに基づいているので、主モードとして音声対話を設定するのが適している。音声対話記述言語には、電話音声応答システムの開発で広く用いられている VoiceXML[9]を用いる。一方、副モードである文字・画像・動画像などは HTML(およびそこから参照されるものとして)で記述する。

3.3 マルチモーダル対話制御フレームの例

図2にマルチモーダル対話制御を行うフレームの例を示す。

宿泊予約 [ID:5] [PID:3]		vxml	html	
Slot名	値	Before Process		After Process
宿泊予定日	[] ②→[2/19]	vxml	html	③
④ 個人情報	[]	個人情報 フレーム を作成		

図2 マルチモーダル対話制御を行うフレーム

例えば、観光案内システムを想定した対話において、現在の焦点がこの宿泊予約フレームに移ったと仮定する。まず、このフレームに付属している HTML ファイルが提示され、次に VXML ファイルによって音声対話が始まる。フレームに付属した VXML ファイルは、この話題に関する一般的なことをユーザに提示するのに用いられる。ここでは、フレームに遷移したことを示す発話を行うとする。そして、最初のスロットである「宿泊予定日」に焦点が移り、以下の手順に従って処理が行われる。

1. このスロットの before Process に VXML ファイルや HTML ファイルが指定されている場合は、これらを実行する。
2. この場合、VXML ファイルでは「宿泊予定日を入力してください」とような発話を行い、日付を認識する文法によってユーザの音声認識しようとする。一方、HTML ファイルはカレンダーを表示し、日付部分のクリックによって宿泊予定日を取得する。
3. これらのどちらかの方法でこのスロットに値が得られた場合、After Process が実行される。
4. この場合は記述がないので、デフォルト処理の次の値が埋まっていないスロットへ焦点を移す。

4. 記述方式の評価

提案したマルチモーダル記述方式の評価として、タイプの異なる複数のタスクを実装して、その一般性を実証した。タスクのタイプは[10]で行った分析に基づき、表1に示す情報の流れる方向の異なる3つのタスクを選択した。

表1 情報の流れる方向に基づくタスクの分類

クラス	情報の流れ	タスク
スロットフィリング	ユーザ→システム	ホテル予約
データベース検索	ユーザ⇄システム	ホテル検索
説明	ユーザ←システム	アクセス情報

例として、ホテル検索対話の一部を説明する。ホテル検索対話は図3に示すようなフレーム構造からなっている。ホテル検索フレームのスロットはエリア・タイプ・施設候補の3つであり、施設候補スロットは子フレームを持ち、その子フレームは一つが選択されれば話題が終了する OR タイプのフレームである。



図3 ホテル検索対話のフレーム構造

まず、ホテル検索フレームの HTML ファイルが起動し、図4に示すような画面表示を行うとともに、図5に示す VXML ファイルを用いて、入力を促す。

ホテル案内システム

本システムは、ホテルを検索し、予約・行程案内を行います。
まずは予約を行います。ホテルの希望エリアとタイプを選択してください。

図4 ホテル検索対話の初期画面

```

<vxml>
  <form>
    <initial>
      <prompt>
        ご希望の地域とタイプを選んでください
      </prompt>
      <grammar src="search.gram"/>
    </initial>
    <field name="area">
      ...
    </field>
    <field name="type">
      ...
    </field>
    <filled>
      <submitnext="server.cgi"/>
    </filled>
  </form>
</vxml>

```

図5 最初のやりとりを行う VoiceXML ファイル

ここでシステムのプロンプト「ご希望の地域とタイプを選んでください」に対して、ユーザが「洛北」と発話すると、次にタイプを求める対話に移る。また、ユーザは画面中のラジオボタンをクリックしても同様の内容を入力できる。

5. おわりに

本稿ではフレーム駆動による対話制御方式を用いたマルチモーダル対話の記述方式を提案した。処理系は Java 言語を用いて実装し、VoiceXML 処理系には Nuance Voice Web Server 日本語版を用いた。その上でいくつかの典型的なマルチモーダルアプリケーションが動作することを確認した。

現在用いている VoiceXML 処理系では、ユーザ発話待ちの状態では外部からの割り込みが行えないため、ポインティングイベントで対話を進めることができない。実装上は、ポインティング入力後 DTMF 入力で、やりとりの終了を知らせている。今後は外部からの割り込みが可能な VoiceXML 処理系を実装することによって、完全に VoiceXML と HTML を並列に扱うことを計画している。

参考文献

- [1] SALT forum: Speech Application Language Tags (SALT) 1.0 Specification, 2002. <http://www.saltforum.org/>
- [2] Jonny Axelsson et al.: XHTML+Voice

- Profile 1.0, W3C Note 2001. <http://www.w3.org/TR/xhtml+voice/>
- [3] T. Nitta, K. Katsurada, H. Yamada, Y. Nakamura, and S. Kobayashi: XISL: An Attempt to Separate Multimodal Interactions from XML Contents, Proc. of EUROSPEECH 2001, pp.1197-1200, 2001.
- [4] D.G. Bobrow, R.M. Kaplan, M. Kay, D.A. Norman, H. Thompson, and T. Winograd, "GUS:a frame-driven dialog system," Artificial Intelligence, vol.8, pp.155-173, 1977.
- [5] S.J. Young, and C.E. Proctor, "The design and implementation of dialogue control in voice operated database inquiry systems," Computer Speech and Language, vol.13, no.4, pp.329-353, 1989.
- [6] D. Goddeau, H. Meng, J. Poliformi, S. Seneff, and S. Busayapongchai, "A form-based dialogue manager for spoken language applications," Proc. of ICSLP'96, pp.701-704, 1996.
- [7] Y. Niimi, T. Oku, T. Nishimoto, and M. Araki, "A rule based approach to extraction of topic and dialog acts in a spoken dialog system," Proc. of EUROSPEECH2001, vol.3, pp.2185-2188, 2001.
- [8] M.K. Smith, D. McGuinness, R. Volz, C. Welty, "Web Ontology Language (OWL) Guide Version 1.0", 2002. <http://www.w3.org/TR/owl-guide/>
- [9] VoiceXML Forum: Voice eXtensible Markup Language (VoiceXML) version 1.0, 2000. <http://www.w3.org/TR/voicexml/>
- [10] M. Araki, K. Komotani, T. Hirata, and S. Doshita, "A dialogue library for task-oriented spoken dialogue systems," Proc. of IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pp1-7, 1999.