

# フレームを用いた VoiceXML における対話制御

## A Frame-based Dialogue Management Method of VoiceXML

荒木 雅弘  
Masahiro Araki

秋田 祥史  
Masashi Akita

京都工芸繊維大学 工芸学部 電子情報工学科  
Dept. of Electronics and Information Science, Kyoto Institute of Technology

In this paper, we propose a frame-driven multimodal dialog control method. By assigning interaction descriptions (voice and visual mode) to a frame or to its slots, the dialogue control description is easy to maintain and to transform to another task domain. The merits of proposed method are (1) realization of dialogue control only by representing task structure by a frame and (2) usage of standard interaction description languages (VoiceXML and HTML) without multimodal extensions. We implemented a frame-based dialogue controller and examined it working in typical multimodal applications.

### 1. はじめに

本稿ではフレームのスロットに VoiceXML および HTML ファイルを対応付けることによって、VoiceXML を拡張することなくマルチモーダル対話履歴の管理を行う手法を提案する。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を実現できることである。

### 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 [Bobrow 1977]、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 [Young 1989], [Goddeau 1996]、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 [Niimi 2001]。

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。

この方式自身はタスク・ドメインに対して汎用性を持つものの、実装言語に Prolog を用いており、構文知識・意味解析知識・対話制御規則・ドメイン依存情報などに可読性・相互依存性などの問題があった。

### 3. VoiceXML を用いたフレーム駆動対話制御方式

2章で説明した方式をより汎用的に利用するために、標準的な知識表現言語と対話記述言語を組み合わせ、もとの言語仕様にできるだけ変更を加えない形でマルチモーダル対話制御を実現することを試みた。

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 [Smith 2002] 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 [McGlashan 2003] 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。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		vxml	html
[ID:5] [PID:3]			
Slot名	値	Before Process	After Process
宿泊予定日	[ ]	vxml	html
	② → [2/19]	①	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 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下さい」のような発話を行い、日付を認識する文法によって認識を行う。一方、HTML ファイルはカレンダーを表示し、日付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. After Process に記述がないので、デフォルト処理として、値が埋まっていない次のスロットへ焦点を移す。

#### 4. 対話システムの実現例

実現例として、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、これらの全てが埋まれば対話が終了する AND タイプのフレームとして実現する。また、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。

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



図3 ホテル検索対話のフレーム構造

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

図4 最初のやりとりを行う VoiceXML ファイル

VoiceXML または HTML から入力された値は、フレームでの対話制御を管理しているサーバに渡され、3章で示した手順で対話が進行する。

#### 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。

#### 参考文献

[Bobrow 1977] D.G. Bobrow, et.al: "GUS:a frame-driven dialog system," Artificial Intelligence, vol.8, pp.155-173, 1977.

[Young 1989] S.J. Young, and C.E. Proctor, "The design and implementation of dialogre control in voice operated database inquiry systems," Computer Speech and Language, vol.13, no.4, pp.329--353, 1989.

[Goddeau 1996] D. Goddeau, et.al: "A form-based dialogue manager for spoken language applications," Proc. of ICSLP'96, pp.701-704, 1996.

[Niimi 2001] 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.

[Smith 2002] 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/>

[McGlashan 2003] Scott McGlashan, et. al.: Voice eXtensible Markup Language (VoiceXML) version 2.0, W3C Candidate Recommendation, 2003. <http://www.w3.org/TR/voicexml20/>