

タスクに依存しないフレーム駆動型対話制御方式

奥 智岐 西本卓也 荒木雅弘 新美康永

京都工芸繊維大学工芸学部電子情報工学科
〒 606-0962 京都市左京区松ヶ崎御所海道町
Email tomoki@vox.dj.kit.ac.jp

あらまし 本稿では、タスクに依存しない対話制御方式について報告する。これは元来テキストベースの対話システムに用いられていたフレーム駆動型の対話制御方式を拡張したものである。この方式では、タスクに固有な知識は、データベースとフレームを用いて記述される。フレーム(特に話題フレームという)には定められたタスクで扱うことのできる関連したひとまとまりの話題と、その優先順位、各話題を終了した時のシステムの行動などが記述されている。話題間に階層関係があるときは、これをフレーム間の階層関係として記述する。対話の進行に伴う対話履歴は、話題フレームの木構造(これを動的話題木とよぶ)として管理される。対話制御部は、この動的話題木を trace することによりシステム発話を決定したり、ユーザの発話の解析を行う。対話制御部を Prolog で実装し、三つの異なるタスクについて良好に動作することを確認した。

キーワード 対話制御, タスク独立, フレーム駆動型, 話題フレーム, 動的話題木

A Frame-driven Task-independent Dialog Control

Tomoki OKU, Takuya NISHIMOTO, Masahiro ARAKI, Yasuhisa NIIMI

Dept. of Electronics and Information Science
Faculty of Engineering and Design
Kyoto Institute of Technology

Matsugasaki, Sakyo-ku, Kyoto 606-0962, JAPAN
Email tomoki@vox.dj.kit.ac.jp

Abstract This paper describes a task-independent dialog control method. It is an extension of the frame-driven dialog control scheme which was originally used in a text-based dialog system. In our scheme, task-dependent knowledge is given as a database and a set of frames (called topic frames). Topic frames are used to describe a set of mutually related topics, their priority, and how the dialog system should act after a topic is closed. Dialog history is represented by a tree of which nodes are topic frames. We call this tree a dynamic topic tree. The dialog controller decides system's actions and interprets user's utterances by tracing the dynamic topic tree. The dialog controller based on this scheme could manage three different tasks.

Key words dialog control, task-independent, frame-driven, topic frame, dynamic topic tree

1 はじめに

近年に音声認識の技術が発達すると共に、その応用である音声対話システムの研究が盛んに行われている。しかし、我々の日常生活にはまだ普及していないのも事実である。その原因の一つとして、同じ対話システムでもタスクを他のものに変更することがコスト並びに労力のいる作業であるということが挙げ

られる。

一般に、音声対話システムの中で最もタスクに依存して設計されているのは対話制御部であり、この部分に関してこれまでも様々な研究がなされてきた。初期の研究では、特定のタスクドメインにおける対話の進行を直接コーディングする方法がとられた[1]。この方法は考えているドメインではうまくいくが拡張性に乏しい。次によく用いられている方法は、

対話の展開を有向グラフで記述するもので [2][3], グラフの節点にシステムが行う質問やシステムの行動が記述され, グラフのエッジには想定されるユーザの応答が記述される. この方法の利点は設計が容易であり, システムの行動の透明性も高いが, 対話システム主導になりがちである.

これらの欠点を克服するために, すなわち拡張性に富み, 主導権交代型の対話の制御が比較的簡単に記述できる方法としてフレーム駆動型の対話制御方式が用いられる. これは最初テキストベースの対話システム [4] で用いられたが, 最近では音声対話システムでも用いられている [5][6]. しかし, 適用されたタスクはいずれも単純なタスクである.

よく知られているように, 目的指向型の対話では, タスクに依存した木構造に従って対話が展開される [7]. 我々はこの事実を利用して, フレーム駆動型の対話制御方式をより大規模なタスクに適用できるように拡張した [8].

本稿ではこの方式をいろいろなタスクに適用して, ある範囲内のタスクではあるが, それがタスク独立に機能することを示す.

2 音声対話システムの概要

本稿における音声対話システムの構成を図 1 に示す. システムは音声認識部, 言語解析部, 対話制御部, 応答文生成部, 音声合成部の五つのモジュールから構成される.

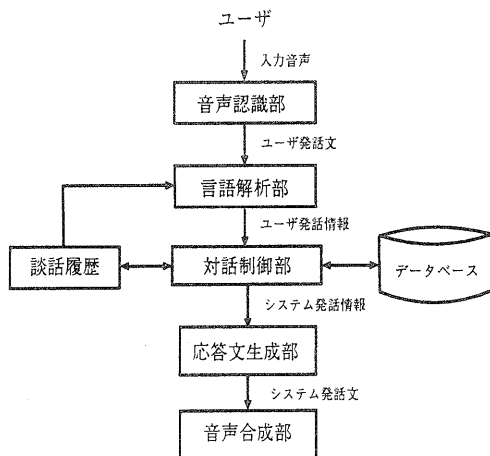


図 1: 音声対話システムの構成

音声認識部で認識されたユーザ発話文の単語列は言語解析部に送られ, 構文解析および意味解析 (図 5 ~ 図 7 参照) が行われ, ユーザ発話情報として話題, 焦点, 談話目標が抽出される [9]. このユーザ発話情報は対話制御部に入力されて談話履歴として記憶されると共に, 対話システムの行動を決定する. システムの応答は, ユーザ発話情報と同形式のもので, 応答文

生成部に送られ, 発話文が生成される.

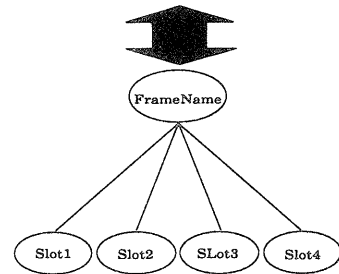
3 フレーム駆動型対話制御方式

3.1 話題フレーム

本稿で提案する対話制御方式は, 人工知能の分野でよく用いられているフレームという知識の表現法を利用したものである. この節ではフレームについて説明する.

Frame Name	State	FrameID number		P_FrameID number
Slot1	Value1	Before Process1	After Process1	Priority1
Slot2	Value2	Before Process2	After Process2	Priority2
Slot3	Value3	Before Process3	After Process3	Priority3
Slot4	Value4	Before Process4	After Process4	Priority4

(a) 話題フレーム



(b) 話題木

図 2: 話題フレームと話題木

フレームは図 2(a) のような構造になっている. 本稿では対話制御に用いるために話題フレームと呼んでいる. フレームにはそれぞれ名前がついており, 固有の ID 番号も付けられている. また, いくつかのスロットがあり, 同じ階層で展開される話題を扱うことができる. 各スロットではそれぞれ値を持つことができ, ここに対話に関するデータが格納される. スロットにはこの値を求めるための処理とそれを求めた後の処理, また処理の優先順位が記述されている.

このフレームは図 2(b) のようにフレーム自体を上位話題, 各スロットを下位話題とした木構造に相当すると考えられる. 話題フレームのスロット値として, 他の話題フレームをとる事を許しているため, 話題フレーム全体は大きな木構造を構成している.

3.2 フレームによる対話制御

ここでは, 図 3 のような京都観光案内タスクの中の観光期間スロットを例にあげて, フレームを用いた対話制御の方法を説明する.

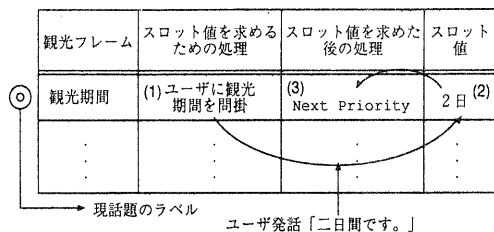


図3：観光期間スロットでのフレームの動き

まず、観光期間スロットに処理が移ると、そのスロットが現話題であることを示すためにラベルが付けられる。そして、スロット値を埋めるために、ユーザに観光期間を問診してその返答をスロット値とする(1)。図の場合、「2日間です。」などといった返答をしたと想定し、この返答よりスロット値は2日となる(2)。その処理が終了と、スロット値を求めた後の処理として、「Next Priority」が行なわれる(3)。これは、同じフレーム内で次に優先順位の高いスロットへ処理、及び話題が移動することを示している。

4 対話制御としてのフレームの動き

4.1 動的話題木と静的話題木

前節で説明したように、話題フレームは木構造に相当すると考えられる。この話題フレームはお互いにリンクを張ることができ、これによって階層構造として展開される話題を扱うことを可能としている。リンクの関係はフレーム固有のID番号によって管理されており、子フレーム(下位話題)にはその親フレーム(上位話題)のID番号が記述されている。また、システムが持っている話題フレーム全体は互いにリンクを張ることができ、1つの木にすることができる。これを静的話題木と呼ぶ。これに対し、実際の対話中に話題が展開することで生成される話題木は動的話題木と呼ばれる。この動的話題木は静的話題木の部分木となっていることは明らかである。システムは対話中に動的話題木を生成することでユーザとの対話の内容、つまり談話履歴を管理している。

4.2 対話の進め方

対話制御システムでは図4(a)のような動的話題木を生成しながら対話を進めていく。対話が進行してある程度の階層までいくとそこから下の階層はデータベースになっており、システム側はデータベース検索を行い、具体的な事例(例えば庭園の特徴や社寺の建立者など)をユーザに提示する。ここではシステムが主導権をとった場合とユーザが主導権をとった場合の対話の進め方について説明する。

システム側は図4(b)のようになるべく早くユーザに具体的事例を提示するように対話を進める。つまり

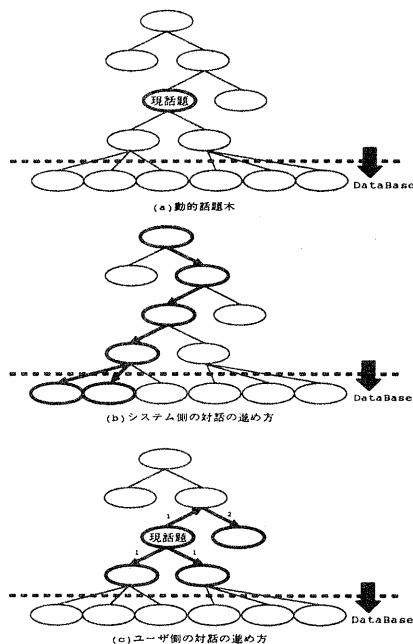


図4：対話の進め方

動的話題木のなかで、焦点とラベル付けされた話題から始めてまだ埋まっていない優先度の一番高いスロットを埋めようと努力する。基本方針としては深さ優先探索(depth-first search)の方針をとりながら対話を行う。

これに対してユーザ発話の解析では、ユーザは話題階層の上位、下位といったことには関係なく、現話題の近傍のことについてしゃべれるものと仮定する。つまり、図4(c)のように現話題(ラベル付けされた枝)からの等距離探索を基本方針としてユーザ発話に含まれている話題に適應するスロットを検索する。このようにユーザが決まったことしかししゃべれないのではなく、現話題の近傍のことをしゃべれることでシステム側とユーザ側の主導権交代型の対話システムが可能となる。

これまで述べたように、本方式ではタスクに依存した情報はすべてフレームに記述している。つまり、この方式であれば、タスクを別のものに変更するときには、基本的に知識データベースとフレームを新たに構築するだけでよい。従って、本方式がタスクに依存しない対話制御方式として有効であると考えられる。

5 フレームハンドラの機能

本稿で提案するフレーム駆動型の制御方式を実現するためには、タスクに依存した情報はすべて話題フレーム及びデータベースとに記述し、フレームを操作するフレームハンドラはタスク独立である必要が

ある。

本システムでのフレームハンドラは次の様な機能を持っている。

- 動的话题木の生成
ある話題フレームのあるスロットが値として他の話題フレームとるとき、あるいはデータベースからあるレコードを読みだしたとき、これをフレーム形式で動的话题木に接続する。
- 焦点・話題の移動(動的话题木の探索)
4.2で説明した動的话题木の探索法により焦点、話題を移動させる。
- システム発話の生成
4.2で説明した深さ優先探索や、データベース探索によりシステムの発話内容が決定されたとき、これを応答文生成部に渡すための形式を整える。
- ユーザ発話からスロット値を抽出
4.2で説明した等距離探索によってユーザ発話中の情報を埋めるべきスロットが決定したとき、これをスロットに代入する。
- データベース検索
現在のシステムでは、データベースは本格的なものではなく、Prologのassertionで与えられている。

ユーザの発話の結果によって対話の流れが変わるような条件分岐のような動作は、例えばAfter Processの所にデーモンを埋め込むことにより実現している。

6 対話制御部の作成と評価実験

6.1 対話制御部の作成

前節で説明したフレームハンドラ機能を論理型プログラミング言語であるProlog言語の述語として作成した。プロトタイプタスクとして京都観光案内を想定し、データベース、静的フレームなどもすべてProlog言語で作成した。そして、そのフレームの中に作成した述語を直接記述してフレームを作成した。

6.2 評価実験に用いるタスクについて

作成した対話制御システムを用いて実際に対話をして、評価実験を行なった。対話制御部への入力は、ユーザ発話を構文解析、意味解析したユーザ発話情報をリスト形式にしたものとし、出力はシステム発話をテキスト出力したものとした。評価実験では、プロトタイプタスクの京都観光案内、そしてタスク依存性を評価するために、別タスクとして京都の宿泊案内とパソコン構築案内を用意した。各タスクでの対話の流れは以下のようなものと想定した。

- 京都観光案内
ユーザに観光期間、興味のある時代や場所を尋ね、京都の社寺、庭園についての特徴や拝観料、建立者などを紹介する。また紹介をしながら、観光期間中の訪問プランを立てる。
- 京都宿泊案内
旅館またはホテルの紹介を行なう。ホテルの場合は客室の料金の提示まで行なう。また、宿泊先の候補の提示の順は、料金の高い順(ホテルの場合はシングル料金)とし、ユーザはこれに対して安い宿泊先にしたいと考える交渉の対話を想定する。
- パソコン構築案内
ユーザに利用目的を尋ね、それに応じたパーツ(ハード、ディスプレイ;プリンタetc)を決めていく。宿泊案内同様、パーツの提示順は値段の高いものからとする。

6.3 結果と考察

各タスクにおける実験結果を図5～7に示す。

システム発話 : こちらは、京都観光案内システムです。観光、宿泊について、何なりとお尋ね下さい。
I: 「観光案内をお願いします。」
[焦点伝達、観光、案内 : [観光 : []]].
システム発話 : 何日の御予定ですか?
I: 「二日間です。」
[情報伝達、期間、観光 : [期間 : [2]]].
システム発話 : 一日目のコースを決めましょう。どのようなところに興味がありますか?社寺と庭園がござりますが。
I: 「庭園に興味があります。」
[焦点伝達、観光興味、行程 : [観光興味 : [庭園]]].
システム発話 : どのような場所に興味がありますか?
I: 「室町時代の庭園が見たいです。」
[焦点伝達、希望時代、行程 : [希望時代 : [室町時代]]].
システム発話 : 室町時代の庭園には金閣寺、銀閣寺、竜安寺などがあります。
I: 「金閣寺の庭園の特徴を教えてください。」
[情報要求、特徴 : [], 庭園 : [特徴 : [], 名前 : [銀閣寺]]].
システム発話 : 月待山の麓、約二万平方メートルを占める広大な庭園です。錦鏡池を中心とする池泉回遊式の下段と枯山水の上段に大別されます。
I: 「分かりました。」
[認知応答、non、non].
システム発話 : 銀閣寺をコースに加えてよろしいですか?
I: 「はい、お願いします。」
[肯定応答、non、non].
システム発話 : 銀閣寺をコースに加えました。あと二箇所ほど、コースに加えることが可能です。室町時代の庭園には金閣寺、竜安寺、大徳寺大仙院などがあります。
:
:

図5 : 観光案内タスクでの対話結果

システム発話：こちらは、京都観光案内システムです。観光、宿泊について、何なりとお尋ね下さい。
 I: 「宿泊案内をお願いします。」
 【焦点伝達, 宿泊, 案内 : [宿泊 : []]].
 システム発話：ホテルと旅館と、どちらがよろしいですか？
 I: 「旅館をお願いします。」
 【焦点伝達, 宿泊希望, 宿泊 : [宿泊希望 : [旅館]]].
 システム発話：どの辺りの旅館がよろしいですか？
 I: 「洛北方面をお願いします。」
 【焦点伝達, 希望場所, 旅館プラン : [希望場所 : [洛北]]].
 システム発話：では松ヶ崎旅館などはいかがでしょう？
 I: 「料金はいくらですか？」
 【情報要求, 料金, 旅館 : [料金 : [query]]].
 システム発話：一泊一万五千円です。
 I: 「他の旅館を紹介して下さい。」
 【他候補要求, 名前, 旅館 : [名前 : []]].
 システム発話：分かりました。もう少しお安いところによろしく。では岩倉旅館などはいかがでしょう？
 I: 「料金はいくらですか？」
 【情報要求, 料金, 旅館 : [料金 : [query]]].
 システム発話：一泊一万円です。
 I: 「そうですか。」
 【認知応答, non, non].
 システム発話：岩倉旅館に決定してよろしいですか？
 I: 「そうですね。」
 【無頓着応答, non, non].
 システム発話：もう少しお安いところを探してみましようか？
 I: 「はい、お願いします。」
 【肯定応答, non, non].
 システム発話：分かりました。もう少しお安いところによろしく。ではカンパール京都などはいかがでしょう？
 :
 :

図6：宿泊案内タスクでの対話結果

システム発話：こちらは、パソコン構築案内システムです。どのような目的で利用されますか？
 I: 「パソコン通信に利用したいです。」
 【情報伝達, 利用目的, 構築 : [利用目的 : [パソコン通信]]].
 システム発話：他に何か利用目的はありますか？
 I: 「はい。」
 【肯定応答, non, non].
 システム発話：利用目的を教えてください。
 I: 「ゲームにも利用したいです。」
 【情報伝達, 利用目的, 構築 : [利用目的 : [ゲーム]]].
 システム発話：他に何か利用目的はありますか？
 I: 「いいえ、ないです。」
 【否定応答, non, non].
 システム発話：パソコン通信、ゲームの目的のためにはハード、メモリ、ハードディスク、ディスプレイ、モデム、サウンド、ソフトが必要です。それではハードについて決めましょう。では、ハードBなどはいかがでしょう？
 I: 「ハードBの値段はいくらですか？」
 【情報要求, 値段 : [], パーツ : [値段 : [], 提示 : [ハードB]]].
 システム発話：三万五千円です。
 I: 「ほかのものを紹介して下さい。」
 【他候補要求, 提示, パーツ : [提示 : []]].
 システム発話：分かりました。もう少しお安いものによろしく。では、ハードAなどはいかがでしょう？
 I: 「値段はいくらですか？」
 【情報要求, 値段 : [], パーツ : [値段 : [], 提示 : [ハードA]]].
 システム発話：三万円です。
 I: 「では、それにします。」
 【プラン決定, non, non].
 システム発話：ハードをハードAに決定しました。それではメモリについて決めましょう。では、メモリBなどはいかがでしょう？
 :
 :
 :

図7：パソコン構築案内タスクでの対話結果

この結果より、観光案内タスクと他の2つのタスクで予想どおりの対話が展開できたことが確認された。また、宿泊案内タスクにおいては交渉の対話を実現でき、フレームを用いて幅広く対話ができることがわかった。さらに、パソコン構築案内タスクでは、京都観光案内とは全くの別のタスクであるのにフレーム、データベースを構築するだけで容易に対話システムを実現できた。これによって、本稿で提案した方式がタスクに依存しない制御方式として有効であることが証明できた。

しかし現在のシステムでは、システム側は料金や値段の高いものからの提示しかできず、交渉の対話としてのまだまだ一般性に乏しいという問題点も挙げられる。そこで今後の課題としては、交渉の戦略をさらに一般化し、それをフレームハンドラの機能として拡張できないか検討することが考えられる。例えば、パソコン構築案内タスクでは、パーツごとに値段の交渉をせず、構築するパーツ全体の予算をユーザ側が提示するといった場面が想定され、また宿泊案内タスクでは、ユーザが最初から一番安い宿泊先を要求する場面などが想定される。こういった場合における提示された条件の言語解析の仕方や、その条件をもとにしてどのようにデータベース検索で扱うかといったことが問題となる。

7 おわりに

本稿では、音声対話システムにおいて現在使用しているタスクから他のタスクへ移行する際にかかる労力、コストを減らすという視点から、タスクに依存しないフレーム駆動型の対話制御方式を提案した。フレームハンドラの機能を Prolog 言語により作成して京都観光案内をプロトタイプタスクとして対話制御部を構築し、その評価を行った。この結果、模擬対話と同じような対話を実現でき、タスクの達成に至った。また、テストタスクとして宿泊案内及びパソコン構築案内で対話実験を行い、本システムのタスク依存性の評価を行なった。これらのタスクではシステムとユーザに違った考え方を持たせ、交渉の対話をするよう設定した。この結果、交渉の対話をしながらタスクの達成に至ることができ、本提案方式がタスクによらない対話制御方式として有効であることを証明した。

今後の課題としては、交渉の戦略を一般化し、フレーム処理として扱えるようなシステムの実現を目指すことを考えている。現在のシステムではユーザが具体的な条件を提示するような対話ができおらず、多少システム主導型の対話になる部分が見受けられる。今後はこの交渉の戦略をさらに一般化し、またユーザが具体的な条件を提示できるようなシステムの構築することが課題である。この課題を達成するにはまず、フレームハンドラ機能の拡張などをしてフレーム処理を見直す必要もあると考えられる。また、ユーザからシステムへの条件提示に関しては、言語解析部からどのような形式でユーザ発話情報が対話処

理部へ伝達されるか、そしてその情報をもとににかにデータベース検索を行なうかが問題になると考えている。

参考文献

- [1] Zue, V., Seneff, S., Glass, J., Goddeau, D., Goddin, D., Pao, C., Phillips, M., and Porifroni, J., "PEGASUS: A Spoken Dialogue Interface for On-Line Air Travel Planning," *Speech Communications*, Vol.15, pp.331-340 (1994).
- [2] Nielsen, P.B. and Baekgaard, A., "Experience with a Dialogue Description Formalism for Realistic Applications," *Proc. of ICSLP'92*, pp.719-722 (1992).
- [3] Zeigler, B. and Mazor, B., "Dialog Design for a Speech-interactive Automation System," *Proc. of EUROSPEECH'95*, pp.113-116 (1995).
- [4] Bobrow, J.G., Kaplan, R., Kay, M., Morman, D., Thompson, H., and Winograd, T., "GUS: a frame-driven dialog system," *Artificial Intelligence*, vol.8, pp.155-173 (1977).
- [5] Young, S.J. and Proctor, C.E., "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] Godden, D., Meng, H., Polifroni, J., Seneff, S., and Busayapongchai, S., "A Frame-Based Dialogue Manager for Spoken Language Applications," *Proc. of ISCLP'96*, pp.701-704 (1996).
- [7] Grosz, J.B. "Discourse Knowledge," in Walker, D.E. eds, *Understanding Spoken Language*, 229-337, North-Holland, New York, (1978).
- [8] Niimi, Y., Takinaga, N., and Nishimoto, T., "Dialogue Management in a Spoken Dialogue System, SDSKIT-3," *Int. Workshop on Speech and Computer (SPECOM'98)*, pp.91-96 (1998).
- [9] Niimi, Y., Takinaga, N., and Nishimoto, T., "Extraction of the Dialog Act and the Topic from Utterances in a Spoken Dialogue System," to appear in *Proc. of ICSLP'98* (1998).