

VoiceXMLの動的生成に基づく自然言語音声対話システム

安達 史博 河原 達也 奥乃 博 岡本 隆志 † 中嶋 宏 ‡

京都大学 情報学研究科

〒 606-8501 京都市左京区吉田本町

† 関西デジタルソフト株式会社

‡ オムロン株式会社

e-mail: adachi@kuis.kyoto-u.ac.jp

あらまし

電話音声入力により、Web から情報を取得するシステムにおいて自然言語入力及び協調的な対話を実現する。VoiceXML であらかじめ静的に対話の流れを記述する枠組みでこれを行うには、種々の入力に応じた VoiceXML を大量に記述しなければならず、また、固定的に対話が記述されていると、検索結果に応じて応答を協調的に生成することが困難である。そこで、CGI スクリプトによって VoiceXML を動的に生成する枠組みを導入した。これに基づいて、京都市バス運行情報案内をタスクとした電話音声対話システムの構築を行った。

Natural Language Spoken Dialogue System based on Dynamic Generation of VoiceXML Scripts

Fumihiro ADACHI Tatsuya KAWAHARA Hiroshi G. OKUNO
Takashi OKAMOTO † Hiroshi NAKAJIMA ‡

School of Informatics, Kyoto University, Kyoto 606-8501, Japan

† Kansai DigitalSoft Co.,Ltd.

‡ OMRON Corporation

e-mail: adachi@kuis.kyoto-u.ac.jp

Abstract

We realize a telephone-based collaborative natural language dialogue system for Web-based information retrieval. Since natural language involves huge number of keyword combinations, a number of VoiceXML scripts need to be created to handle all possible input patterns. If dialogue is fixed beforehand, it is not easy to generate collaborative responses from the query results. In this paper, we present a technique to generate VoiceXML scripts dynamically using a CGI script. Based on this framework, we have developed a telephone-based dialogue system to access Kyoto city bus information.

1 はじめに

近年の音声認識技術の向上により、その技術を用いた音声対話システムの研究が活発に行われており、対面型の対話システムに限らず、電話を用いた対話システムの研究も多く行われている [1, 2, 3]。このようなシステムの研究開発においては、実際に動機づけられたユーザによる評価が重要であり、一般ユーザが積極的にシステムを使える環境を構築することが望ましい。

一方、電話による音声入力により、Web にアクセスし情報を提供するボイスポータルシステムが米国では2年ほど前から実用化され、日本でも昨年よりVoizi[4]などでサービスが開始されている。このようなアプリケーションの普及とポータビリティの維持のために、標準的な記述言語としてVoiceXML[5]が策定され、VoiziもVoiceXMLを使用して構築されている。しかし、VoiceXMLによって固定的に対話を記述すると、多くの場合、システム主導・単語認識ベースのシステムに限定される。システム主導による対話は、ユーザがシステムの要求するスロットに単語を埋めることで対話が進められるが、必須でないスロットが存在するタスクにおいては実現が難しく、適用できるタスクが限定される。また、このような固定的なインターフェースは確実性の点で優れているが、必ずしも効率的、ユーザフレンドリとはいえない。

そこで、本研究ではVoiceXMLを用いた対話システムにおいて、混合主導でかつ自然言語による入力を行えるようにする。自然言語のように入力が多様化すると、全てのパターンに応じた処理を記述するのは困難となる。本研究では、VoiceXMLを動的に生成する枠組みを導入することでこれを実現した。また、その枠組みに基づいて、バス運行情報案内をタスクとした電話音声対話システムを構築した。

以下、2章でVoiceXMLの枠組みで自然言語入力・協調的応答を実現する方法を説明する。3章では具体的にタスクを設定して実装した電話音声対話システムについて説明し、4章で協調的な応答の生成について述べる。5章で本システムにおける動作例を示し、6章で本稿のまとめを行う。

2 VoiceXML における柔軟な対話の実現

VoiceXMLは、主に電話による音声対話における対話フローを記述するための言語であり、ボイスアプリケーションの標準言語である [6, 7]。VoiceXMLは通常図1のように、(1) 合成音声として出力する内容の指定、(2) ユーザ発話を受理する文法の指定、(3) 対話のプランニングの記述の3つの部分からなる。

```
<?xml version="1.0" >
<form id="shop">
  <block>いらっしゃいませ。パソコンショップへようこそ。</block>
  <field name="start">
    <prompt>ノートパソコンかデスクトップを選択してください。</prompt>
    <grammar>ノートパソコン|デスクトップ</grammar>
    <filled>
      <if cond="start == 'ノートパソコン'" next="./note.vxml"/>
      <if cond="start == 'デスクトップ'" next="./desktop.vxml"/>
    </filled>
  </field>
</form>
</vxml>
```

図 1: VoiceXML の記述例

現在、VoiceXMLを用いてWeb上の情報を取得できるサービスの多くでは、システム主導によりユーザの発話を制限したり、単語単位での入力しか受理しないようになっている。しかし、Web上に存在する大量の情報から自分の欲する情報を効率良く取得するには、自然言語によって直接的に指定できることが望ましい。また、自然言語による対話システムでは、システムとユーザが協調的に対話を進めていくことが重要である。

2.1 自然言語入力の実現

音声対話システムにおいて、ユーザが自由に発話できることは重要である。しかし、自然言語発声では多数のキーワードが任意に含められるので、その組み合わせが膨大になる。VoiceXMLにおいて、このような多様な各入力に応じた処理をあらかじめ静的に記述するのは困難である。例えば図1では、入力が2種類に限定されているため、対話管理部分である(3)では2種類の処理を記述すればよいが、1文で複数のスロットやキーワードが指定される可能性を許すと組み合わせ的に増大する。また、全て用意するのは語彙の変更などに対するポータビリティの点で問題である。

そこで、VoiceXMLを動的に生成する枠組みを導

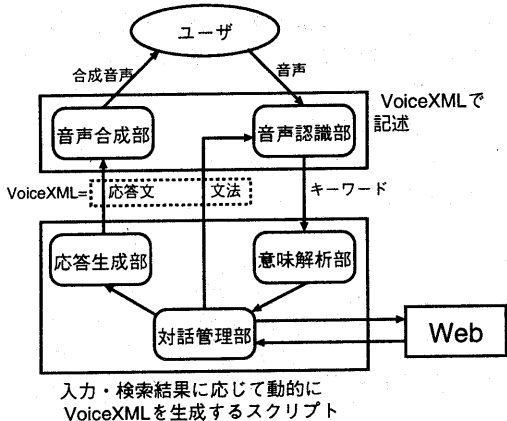


図 2: VoiceXML の動的生成による音声対話システムの概念図

入することで自然言語入力を実現する。このシステム の概念図を図 2 に示す。つまり、ユーザとのインタラクション部分は VoiceXML が役割を担い、対話のプランニング部分は VoiceXML から分離する。

VoiceXML で指定した文法を用いてユーザの発話は認識され、認識結果もしくは認識結果から抽出したキーワード群が VoiceXML 動的生成スクリプトに渡される。VoiceXML 動的生成スクリプトは、その入力に応じて Web を検索し、応答用 VoiceXML を生成する。他にも、キーワードの組み合わせに矛盾が生じればそれを指摘する VoiceXML を、入力に不足があればユーザ発話を誘導する VoiceXML を、入力に曖昧性があればそれを解消する VoiceXML を生成する。

これにより、多様な入力に対して大量に VoiceXML を記述する必要がなく、また、システムの変更も容易に行えるポータビリティが実現できる。

2.2 協調的な応答の実現

音声対話システムにおいて、ユーザの意図を推察して協調的な応答を生成することは重要である。これには、ユーザの要求に対する答え以上に情報を付加したり、検索が失敗したらその理由を伝えたり、代替案を提示することが含まれる。ユーザの発話だけでなく検索結果などの状況に応じて応答文を協調的に変化させるには、検索結果の内容を解釈する必要

がある。

例えば、パソコンの購入における「A 社の 15 万円以下のノート PC。」という入力に対して該当する商品が存在しなければ、「B 社ならあります。」というように提案することを実現するには、全ての入力に対する結果を理解しておく必要があり、静的に VoiceXML を用意して行うのは困難である。特に、時間によって検索結果が変化する場合に困難の度合いが増す。

そこで、Web の検索結果に応じて適切な応答文を生成し、それを VoiceXML の形式に変換して出力する。意味解析・対話管理・応答生成のモジュールを、固定的に記述される VoiceXML から分離することにより、入力条件と検索結果を解析し、それに基づいて協調的な応答を生成することができる。

3 京都市バス運行情報案内システム

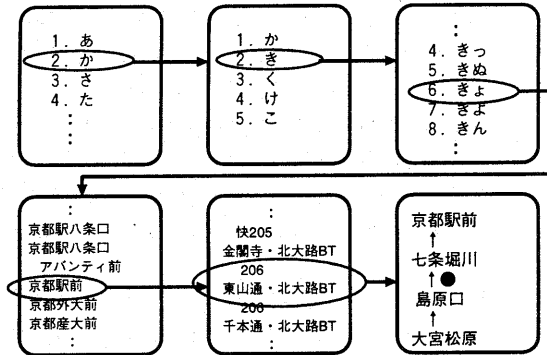
3.1 バス運行情報案内タスク

2 章で述べたような VoiceXML を動的に生成する枠組みに基づいて音声対話システムを構築した。

タスクは京都市バス運行情報案内である。これはユーザの指定するバスが、乗車する停留所の何駅手前まで接近しているかを知らせるシステムである。京都市内のような市街では、バスの運行は時刻表通りに行われず、そもそも「5~10 分おき」と時刻表に書かれている場合もある。運行情報は刻々と変化しており、また、急いでいるユーザにとってはすぐに情報が欲しい状況であり、音声入力による情報検索に適したタスクといえる

本システムでユーザに提供するバスの接近情報は、ポケロケ(携帯型バスロケーションシステム)[8]から取得する。ポケロケは現在、インターネット、i-mode などの携帯電話で利用が可能であり、1 日あたり約 1 万件のアクセスがある。携帯電話でポケロケを利用する際の動作例を図 3 に示す。例えば、乗車駅である「京都駅前」を指定するのに、「か」→「き」→「きよ」→「京都駅前」と選択し、系統番号の一覧から目的のバスを選択することで、接近情報が得られる。

しかし、このように携帯電話でバスの接近情報を取得するには、繰り返しデータを送受信することとなり、また、表示画面も小さいため頻繁にスクロー



京都駅前前で206系統の東山通・北大路BT行きのバスに乗車する場合

図 3: ポケロケの動作例

ルを行う必要がある。そのため、ユーザは接近情報を素早く取得したいにも関わらず操作が煩わしい。また、そもそも行き先はわかっている場合でもバスの系統番号は知らない場合もある。したがって、音声言語によって直接的に乗車するバスを指定できれば非常に効率がよい。

3.2 システムで扱う語彙

本システムで扱う語彙の種類は以下の3つである。

- 乗車するバスの停留所の名称 (START)…必須項目
- 降車するバスの停留所の名称 (GOAL)
- バスの系統 (ROUTE)

図3に示すように、バスの接近情報を取得するには、乗車するバスの停留所・バスの系統・バスの行き先の3項目を特定する必要があるが、「東山通・北大路BT行き」といったバスの行き先を正確に発話できる人はほとんどいないと判断し、これはSTARTとGOALの組み合わせから推定する。STARTとGOALで複数の経路がある場合は、停留所により接近している系統・START-GOAL間の停留所数がより少ない系統の接近情報を優先的に伝える。ROUTEのみが指定された場合は、2通りの方向があるので両方の接近情報を伝える。また、正確な停留所名以外にも、付近の施設の名称や略称などでも入力できるように語彙を設定した。本システムにおける語彙

を表1に示す。これには、京都市営バスのほぼ全ての停留所・バスの系統が含まれる。本研究では、以下のような発話を想定してシステムの構築を行った。

(発話例)

「百万遍から京都駅まで。」

「四条河原町から203系統のバスに乗ります。」

「清水寺に行きたいのですが。」

表 1: システムの語彙の一覧

停留所の名称の数	652
バスの系統の数	85
付近の施設の名称の数 + 略称の数	756

3.3 システムの実装

VoiceXMLを動的に生成する枠組みに基づいた京都市バス運行情報案内を行う音声対話システムの構成を図4に示す。以下、内部の処理に関して述べる。

(1) Voice Web Server

ボイスポータル用ブラウザ VWS(Voice Web Server)は、VoiceXMLを解釈し、応答文を合成音声として出力し、指定した文法によってユーザの発話を認識する。そして、認識結果から抽出したキーワードを出力する。

(2) VoiceXML Generator

VoiceXML GeneratorはPerlによって記述されたCGIスクリプトである。ユーザの発話した内容やWebから取得した接近情報に応じて、適切な応答を生成し、VoiceXMLの形式に変換する。また、誘導発話や確認発話などを行うVoiceXMLを生成した場合、ユーザの次発話を認識する文法の指定も行う。

(3) 知識ベース

知識ベースはあらかじめ、ポケロケや京都市交通局のWebページより作成したもので、各バスの系統に含まれる停留所のリスト・付近の施設と停留所との対応関係・固有名詞の読み情報などがこれに含まれる。

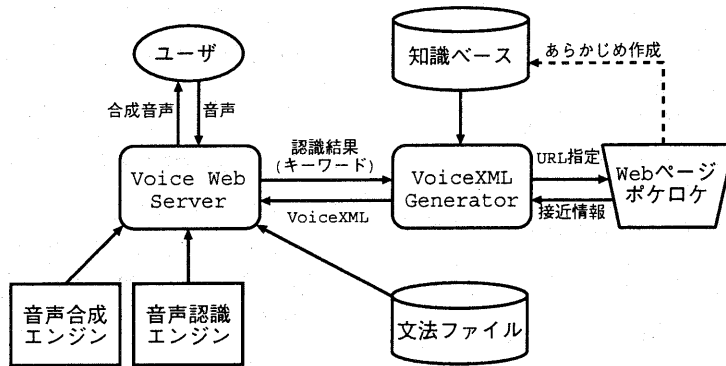


図 4: 京都市バス運行情報案内システムの構成図

4 協調的な応答の生成

動的に VoiceXML を生成することで、Web の検索結果などの状況に応じて、応答文を協調的に変化させることができる。協調的な応答は、ユーザーの要求に対する答え以上の情報を付加する、検索が失敗の場合はその理由を伝える、代替案を提示する、内包的な表現を用いることなどによって実現される [9]。以下のうち、提案的な応答・矯正的な応答は本システムに実装されており、内包的な応答・補完的な応答は現在実装を検討中のものである。

(1) 提案的な応答 (Suggestive answers)

検索結果の内容がユーザーにとって否定的なものであっても、関連のある情報を提供する。

(応答例)

U: 「17 系統で百万遍から京都駅まで。」

S: 「17 系統のバスはしばらく来ませんが、206 系統のバスは 4 駅手前の高野を出発しています。」

(2) 矯正的な応答 (Corrective answers)

検索に失敗した場合や、ユーザーの発話に誤りが含まれている場合にその理由を伝える。

(応答例)

U: 「京都駅から嵐山まで」

S: 「京都駅から嵐山まで直接行くバスはありません。」

(3) 内包的な応答 (Intensional answers)

検索結果が多い場合、内包的な表現にした方が、

ユーザーにとってわかりやすいと考えられる。

(応答例)

U: 「四条河原町から四条烏丸まで。」

S: 「四条河原町に来る 200 番台の系統は全てまもなく到着します。」

(実際には、201,203,207 系統がある。)

(4) 補完的な応答 (Completion answers)

ユーザーの要求以上に情報を含めて応答することは協調的であるが、どのような情報をどのような時に応答に含めるかの検討が必要である。

(応答例)

U: 「京都駅の 5 系統のバス。」

S: 「5 系統のバスは A1 のりばから 17:34 に出発します。」

(京都駅前には乗り場が多い。)

5 対話システムの動作例

本研究で作成した京都市バス運行情報案内システムは、現在、電話インタフェースで動作している。基本的な対話戦略として、ユーザーの自由発話は最初のプロンプトで許容し、誤認識などによりユーザーの発話の意図通りの結果が得られない場合は、システム主導に切り換えることにしている。

本対話システムの動作例を以下に示す。

(動作例 1)

U: 「百万遍から 206 系統のバス」

S: 「206 系統の東山通・京都駅行きバスは 6 駅手

前の高木町を出発しています。」

(動作例 2)：不足項目を尋ねる

U：「17 系統のバスで京都駅に行きます」

S：「どこの停留所でバスに乗車するのですか？」

U：「百万遍です。」

S：「17 系統の河原町通・京都駅行きのバスは 2 駅手前の北白川を出発しています。」

(動作例 3)：曖昧性の解消

U：「四条河原町」

S：「四条河原町からバスに乗車するのですか？」

U：「そうです」

S：「どこの停留所でバスを降りるのですか？」

U：「嵐山です。」

S：「11 系統の嵐山・山越中町行きのバスは、2 駅手前の三条京阪前を出発しています。」

6 まとめと今後の課題

本稿では、ボイスアプリケーションの標準言語である VoiceXML を用いた音声対話システムにおいて、自然言語入力・協調的な応答を実現する方法を検討した。入力が多様化すると VoiceXML の記述量が増え、検索結果が多様化すると、固定的に記述する方法では対処できない。そこで、VoiceXML を動的に生成する枠組みを考え、バス運行情報案内をタスクとした電話音声対話システムを実装した。

現在、一応の動作をしているが、今後は、一般ユーザによるシステムとの対話データを収集し、その評価およびフィードバックをすることで、システムの改善を図り、市民に対して公開していく予定である。また、ユーザモデルを導入することで、個々のユーザに応じた応答を生成することも検討中である。

参考文献

- [1] CMU Communicator.
<http://www.speech.cs.cmu.edu/Communicator/>
- [2] The MIT GALAXY System.
<http://www.sls.lcs.mit.edu/GALAXY.html>

- [3] RAILTEL.
<http://www.limsi.fr/Recherche/TLP/railtel.html>.
- [4] Voizi. <http://www.voizi.net/>
- [5] VoiceXML Forum. <http://www.voicexml.org/>
- [6] 桂田浩一, 中村有作, 山田真, 小林聡, 山田博文, 新田恒雄. 音声対話記述言語 VoiceXML と MMI 記述言語 XISL の比較. 情報処理学会, 2001-SLP-38-8, 2001.
- [7] 鯨井俊宏, 高橋久, 天野明雄, 畑岡信夫. VoiceXML インタプリタと連続単語認識エンジンの開発-音声ポータル向け音声認識技術の開発-. 情報処理学会, 2000-SLP-33-12, 2000.
- [8] ポケロケ (携帯型バスロケーションシステム).
http://www.city.kyoto.jp/kotsu/news/bls/bls_manu_top.htm
- [9] David SADEK. Design Consideration on Dialogue Systems: From Theory to Technology - The Case of Artemis -. Proc. ECSA Workshop on Interactive Dialogue in Multi-modal Systems, Kloster Irsee, 1999, pp.173-187.