

データベースアクセス機能を有する自由発話対話プラットフォーム

甘粕 哲郎[†] 平沢 純一[‡] 富久 昭弘[†] 宮崎 昇[†]

[†] 日本電信電話株式会社
NTT サイバースペース研究所
〒239-0847 神奈川県横須賀市光の丘 1-1

[‡] NTT コミュニケーションズ株式会社
先端 IP アーキテクチャセンター
〒163-1422 東京都新宿区西新宿 3-20-2

概要

本稿では、VoiceXML や我々が提案した CASYSML など、WWW 技術をベースとした音声対話処理記述言語において、知識処理にリレーショナルデータベースを利用する際の、対話システムの開発面からの問題点を検討する。そして、(1)基本検索関数、(2)検索結果の絞り込み機能、(3)検索実行時情報の取得およびエラー処理の自動化(4)同一性判定処理の機能をもつ”DB アクセス仲介モジュール”を音声対話処理記述言語のインタプリタに追加する構成を提案する。

A Platform with RDBMS Access for Spontaneous Spoken Dialog Systems

Tetsuo Amakasu[†], Jun-ichi Hirasawa[‡], Akihiro Tomihisa[†], Noboru Miyazaki[†]

[†] NTT Cyber Space Laboratories, NTT Corporation

1-1 Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847 Japan

[‡] Innovative IP Architecture Center, NTT Communications Corporation

3-20-2 Nishi-Shinjuku, Shinjuku-ku, Tokyo, 163-1421 Japan

Abstract

This report proposes a method for using RDBMS as knowledge-bases in WWW based spoken dialog systems such as VoiceXML and CASYS-ML. We investigate problems in customary way of accessing RDBMS in point of practical system development. And we propose “DB access mediator module” that has (1) Basic search, (2) Additional filtering, (3) Automatic search information acquisition and error-recovery, and (4) Semantic similarity judgment functions.

1. はじめに

VoiceXML1.0 の提案を契機に音声対話システムはアプリケーション技術として普及期の段階を迎え、そしてその適用の幅を広げようとしている。

まず、米国で 2000 年に音声ポータルサービスが

始まった。日本でも、2001 年 8 月からは NTT コミュニケーションズ社 (V ポータル[1]) がサービスを開始している。電話上のシステムは現在も発展を続け、W3C による VoiceXML2.0 の勧告[2]や IP 化される電話網との連携が図られている。

SYS	店舗検索システムです。お探しいエリアとジャンルをおっしゃってください ・複数スロットの同時待ち受けが可能
USR	えーと、それじゃあ、横浜の映画館はどうですか？ ・複数入力項目の同時発声が可能 ・周辺表現や不要後に対応
SYS	エリア横浜、ジャンル映画【館ですね？】 ←出力の途中停止 ・任意箇所での割り込み可能 (フルタイムバージョン)
USR	いや、やっぱり関内をお願いします ・入力済の値の修正/変更が可能
SYS	では、横浜ではなく関内、ジャンルは映画館でよろしいですね？ ・スロット値の変化を把握
USR	はい、お願いします
SYS	それですと、6件になります...
:	:

図 1: 自由発話入力における対話例

さらに、携帯端末やカーナビゲーションシステムなどをターゲットに、音声入出力を含めたマルチモーダル対話コンテンツ記述仕様の提案や標準化の検討が行われている[3]。

NTT サイバースペース研究所では、音声対話システムの実用化において表出した問題点の研究に取り組んできた。一例として、自由発話入力技術を開発し既に報告している[4]。

自由発話入力技術は、これまでの音声対話サービスが「特別な利用法の習得が不要」という音声対話インタフェースの利点を活かしていなかった問題に対処したものである。

特徴としては、クラス N -gram を用いた確率言語モデル音声認識の採用、発話理解機能と対話（文脈）理解機能の強化が図られている。結果、利用者の利便性について以下の点が改善された。

- 1) 制約の緩い自由な発話表現の許容、および言いよどみ等の自然発話中の現象の許容
- 2) 1 ターン分の入力発話中での入力項目（スロット）の数および順序の自由な選択。
- 3) システムガイダンス再生途中の割り込み（バージョン）時に生じるターンの乱れを低減。

この技術による対話例を図 1 に示す。さらに、我々は、この自由発話入力技術をプラットフォームにも実装した。"RexDialog"[4] は電話 IVR (Interactive Voice Response: 自動応答装置) 上に搭載され、サイバリアテンダント[5] (図 2) マルチモーダル入力に対応している。

サイバリアテンダントでは、対話処理記述言語 "CASYS-ML" を提案するとともに、CASYS-ML を解釈実行し音声認識エンジン VoiceRex[7] 等をコントロールするインタプリタを Web ブラウザのアドオンとして実現した。サイバリアテンダントは Web ブラウザ上で動作するため、従来の音声のみを用いる対話システムに比べ、画像やテレビ電話システムと連携できるなど、多彩なシステムが構築可能である[6,8]。

本稿では、音声対話システムプラットフォームをさらに実用的にするために、対話で用いる知識をリレーショナルデータベースシステム (RDBMS) で管

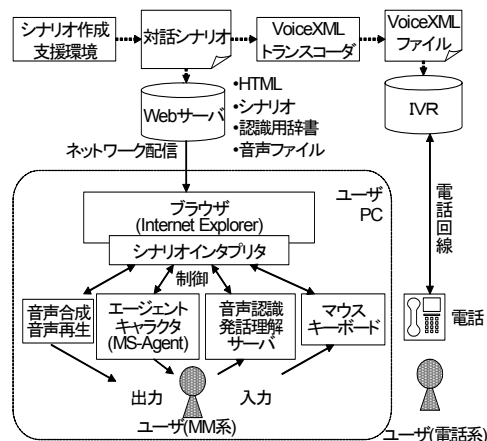


図 2: サイバリアテンダントの構成

理・アクセスする枠組みについて述べる。次節以降、音声対話処理における RDBMS 利用の必要性(2 節)を述べ、現状の手法では対話中の知識処理とデータアクセス処理が未分離である問題について述べ(3 節)、それを解決する一手法を提案する(4 節)。

2. 音声対話処理での RDBMS の利用

本格的な音声対話システムの普及に向けては、有用なサービスを提供することが課題である。有用なサービスとは、ユーザの日常的な活動での問題解決に役立つ情報(コンテンツ)を提供する、もしくは、問題そのものを解決するサービスである。

WWW の分野などで既に広く普及しているサービスでは、コンテンツ上の知識となる情報はデータベース(DB)に保存されている。例えば、ホテル予約であれば空室情報、気象情報案内システムにおいては地方毎の気象情報が保存されている。

DB 側では、サービスが更新される度に新たな属性を追加する等、情報を修正する必要がある。また、ホテル予約システムであれば、同時にアクセスしている他のユーザとの間でダブルブッキングが発生しないように(トランザクション)管理されている必要がある。これらの必要性から、WWW や電話で提供されている多くのシステムでは、内部で利用するデータを RDBMS で管理している。

VoiceXML や CASYS-ML 等 WWW の技術をベースとしている対話記述言語も他の WWW 系のシステムと連携するためには、RDBMS 上の情報にアクセス可能である必要がある。

WWW ベースの対話処理記述言語では、現状でも CGI[11]や JSP[12]などの仕組みを用いれば DB を利用した音声システムを実現することはできる。しかし、実用的なサービスを設計・開発・保守する観点に立てば、機能が不十分である。次にその理由を述べる。

3. 現状の技術と問題点

3.1. 現状の DB アクセス手法

図 3 に現状の音声ブラウザと RDBMS を連携させ

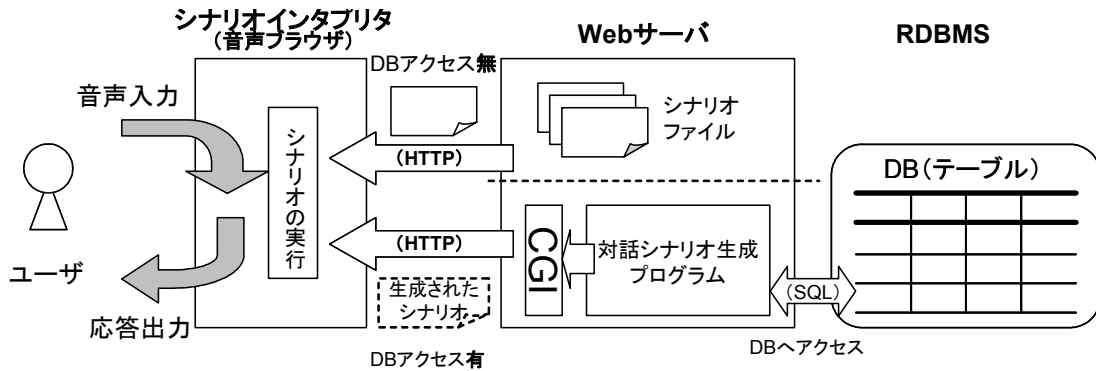


図 3: 従来手法による DB アクセスの構成例

る場合の典型的なシステム構成例[11]を示す。

RDBMS では、2 次元テーブルの形でデータが管理される。ひとつのデータはテーブルの各一行を構成し、各カラム（列）にそのデータを構成するデータ要素が収められている。データを利用するクライアントプログラムやデータベースの維持管理者は SQL などの専用言語を介することでデータにアクセスする。

音声ブラウザから DB にアクセスするには、DB アクセス処理内容をどこかで SQL に変換する必要がある。そこで変換のために、“対話シナリオ生成プログラム”を作成して対処する。この“対話シナリオ生成プログラム”では、まず音声ブラウザからの要求を SQL に変換した上で DB に処理を要求する。その後、その DB 処理の結果が、対話システムの次の応答内容に含まれている、新たな対話シナリオを生成することができる。

対話シナリオ生成プログラムは HTTP サーバの背後に設置され、CGI インタフェース等を介して音声ブラウザと接続される。

3.2. 音声対話システムの知識処理

音声対話システムで知識へのアクセスが行われるのは、ユーザからの入力完了し、対話の結果として検索結果などを応答出力する時のみではない。

図 4 に典型的な音声対話システムのアーキテクチャを示す[9]。この中で例えば語彙に関する知識に着目すると、音声理解部では音声理解規則と各単語の意味属性、品詞等の知識を適用して、一発話中の意味表現を音声理解結果として出力する。また、対話制御部では、次のシステムの行動を決定する際、語彙の親子関係(天気予報の場合:神奈川県には東部、西部の選択肢がある)などの情報が存在するか検査する場合もある。このように、応答出力以外の部位においても知識の利用は必要となる。

3.3. DB アクセス処理における問題

3.1 節で示した DB アクセス処理の手法で DB を知識源とする音声対話サービスを構築しようとするのでは、下記の 2 つの問題が発生する。

【問題1】音声対話シナリオの分散

従来の手法では、音声対話シナリオのうち、DB ア

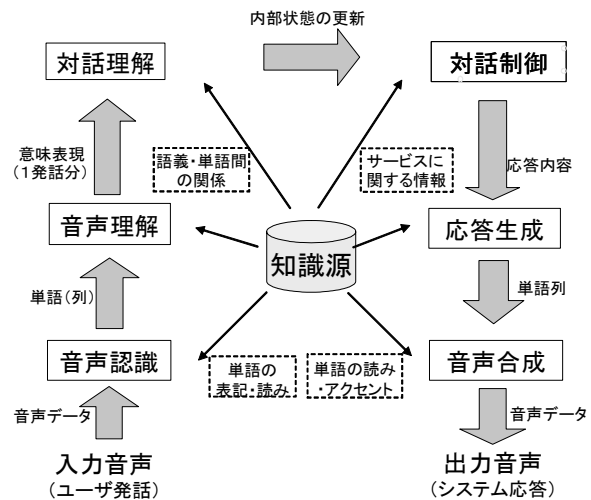


図 4: 対話処理における知識の利用

クセスの必要がなく、利用者とのやり取りの文脈のみで応答が決定される対話シナリオは、静的な XML 文書ファイルとして HTTP サーバ内に保存される。

しかし、DB を参照した結果に基づいて応答が決まる対話シナリオは、DB アクセス処理を行う“対話シナリオ生成プログラム”の中に応答決定過程が埋め込まれてしまう。

つまり、一連の音声対話シナリオの文脈処理に関する処理が文書ファイルと対話シナリオ生成プログラム中に分散されてしまう。その結果、対話シナリオの設計者は対話シナリオを修正するなどのメンテナンスが困難になる。

【問題2】音声対話処理・DB アクセス処理境界の不明確化

対話シナリオ生成プログラムの中では、DB アクセスとその結果に基づいた知識処理を繰り返しながら応答内容を決定し、1 つの対話シナリオファイルが生成される。これは、3.2 節で述べたとおり、対話処理のそれぞれの段階で知識へのアクセスが行われるためである。

そのため、対話シナリオ生成プログラムの中では DB アクセス処理を記述する部分と対話処理を記述する部分の境界が不明確になってしまう。

その結果、データベースアクセス処理の詳細設計

と対話シナリオの詳細設計作業を分担して行うことが難しくなっている。

3.4. 一般的な解決方法とその問題点

上記の問題は、応答文の提示順を決定するロジックと応答文そのものを分離する問題と同じにも見える。例えば、HTML の分野では、Java 言語による技術である Servlet, JSP(Java Server Pages), JavaBeans の組み合わせ(MVC モデル)[12]でこの問題を解決している(図 5)。

MVC モデルでは、個々の場面で用いる HTML の大まかな内容や画面レイアウト(view)はデザイナーがあらかじめ HTML 様のテンプレート(JSP)として記述する。JSP の中で DB 処理の結果に基づいてテキストが変化する箇所では、そのテキストを生成する JavaBeans のクラスメソッドを指定しておく。DB アクセス処理の詳細は JavaBeans により行われる。このモデルは、記述の分担を可能にするのみならず、DB の仕様が変更になった場合でも変更が最小限で済むという利点がある。

しかし、HTML と異なり VoiceXML などの音声対話記述言語を用いる場合、MVC モデルでは不十分である。MVC モデルでは、Servlet はブラウザより送信されたフォーム値を検証し、次に画面に提示する JSP を決定する。JSP からは画面上の静的な表示内容が記述された HTML 文書が生成される。一方、音声対話の場合 HTML 文書に相当する音声対話シナリオの中には、対話制御に関する動的処理の記述が含まれる。ここで対話制御の記述には、複数発話間にまたがる理解処理やユーザ入力値の検証など、ユーザへの応答を決定する動的な処理を含む。もし、MVC モデルの枠組みで対話制御を行おうとすると、対話制御に関する記述が JSP から Servlet に分散し、【問題 1】が依然残ってしまう。

本稿では、3.3 節で述べた 2 つの問題点を解決するために、音声対話処理の中で行われる知識処理において、DB アクセスに関する処理と対話シナリオに関する処理の範囲を明確にし、さらに、DB アクセスの有無に伴って対話シナリオが分散してしまうことを避けるのが課題であると考え、解決策を検討した。

4. 音声対話処理と DB 処理の分離

4.1. 提案手法

3.3 節で述べた課題に対し、本稿では以下のアプローチをとった。

【問題 1】に対して、音声対話シナリオが分離することを防ぐために、音声対話シナリオと DB の間に立って SQL を発行し、DB アクセス処理を行うモジュール(DB アクセス仲介モジュール)を設置する。さらに、このモジュールには次の【問題 2】へのアプローチで抽出する汎用的な処理を音声対話シナリオ側から簡便に呼び出せるインタフェースを設ける。これにより DB アクセスの有無に関わらず、音声対話シナリオを一元的に管理することができる。

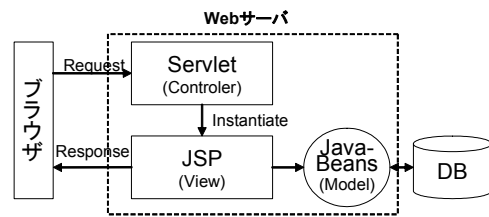


図 5: Java 言語技術による MVC モデル構成

【問題 2】に対して：記述言語中の対話処理と DB アクセス処理の分離を行うために、音声対話処理の際に行われる DB アクセス処理を整理し、汎用的な処理を設定する。これにより DB アクセス処理に関する記述だけを記述言語中から明確に分離することができる。

図 6 に DB アクセス仲介モジュールを設置した本方式による対話プラットフォームの構成を示す。

DB アクセス仲介モジュールに対して音声対話記述言語から呼び出せる汎用的処理機能として以下の 4 機能を設定した。

- ・ 基本検索関数
- ・ 検索結果の絞り込み機能
- ・ 情報取得、エラー処理の自動化
- ・ 同一性判定処理

次節にて各機能の詳細について述べる。

4.2. DB アクセス仲介モジュールの機能

・ 基本検索関数

音声対話シナリオ内で DB 検索処理を実行する場合は、入力された「スロット - スロット値」の組に該当する情報を取得するのが基本的な処理である。そこで、検索結果として取り出すべきデータレコードの条件を「カラム名 比較演算子(=等) 検索条件値」のように、条件判定対象としたい「カラム名」とその値(「検索条件値」)、判定のための「比較演算子」からなる条件式とする。この条件式を複数組み合わせ、対話シナリオの冒頭で定義する。これを基本検索関数と呼ぶ。

ここでは、比較演算子として等号(=)、不等号(!=)、大小比較(<,>)の一般的な演算子の他に、部分文字列マッチングを指定可能とした。等号以外の演算子は、スロット値に対する付加条件に対応し、それぞれ不等号は否定条件、大小比較は範囲指定(「以上」「以下」)、部分文字列比較は「～を含むもの」といった付加条件に対応する。

対話シナリオ中で基本検索関数を利用する際には、基本検索関数名とその中の各条件式において、比較演算子でカラム名と比較する検索条件値を対話の状況に合わせて指示することができる。

・ 検索結果の絞り込み機能

上記の基本検索関数で検索すると、通常、条件に該当する複数の結果が得られてしまう。しかし、得られた検索結果を応答内容としてすべてユーザに提示することが好ましいとは限らない。

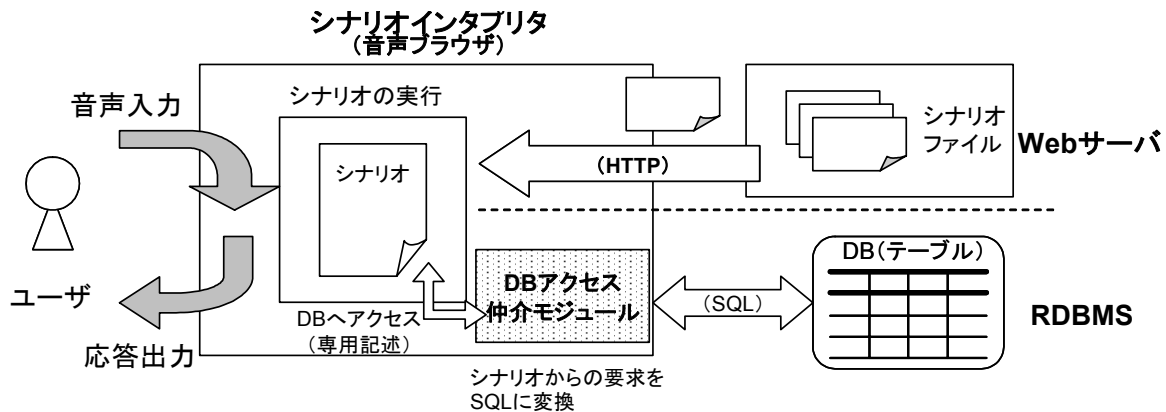


図 6: 本アプローチにおける構成

例えば一回の検索で多くの検索結果が得られてしまった場合、全結果をそのまま音声で読み上げるのは適切な応答とはいえない。その時点までの対話の文脈的な状況やシステム制約から「今は候補の中で一番安いホテルがわかればよい」または「表示・応答の都合上、ユーザに選択肢として与えられるのは3候補以内」などの条件で検索結果の絞り込みをさらにを行い、有望な結果だけを取り出して応答するという処理が有用になる。

上記で示した対話の文脈的な状況やシステムの制約は、基本検索関数の条件式で用いる比較演算子だけでは表現が難しい。

そこで、プラットフォームの機能としてあらかじめ絞り込みの機能を準備した。対話シナリオでは、基本検索関数実行時に”絞り込み指定子”を併用して、この絞り込み方法を指定することができる。今回準備した絞り込み機能の一覧を表1に示す。

・検索実行時情報の取得とエラー処理の自動化

実際に取得するデータの数は絞り込み機能で絞り込めるが、基本検索関数の検索条件に該当する全データ数も対話戦略上重要な情報である。例えば、検索条件に該当するデータの数が非常に多い場合、絞り込み機能で絞り込んだデータをユーザに提示する前に、検索条件自体を変更するようユーザに促す応答を行う方が適切な戦略といえる。

そこで、絞り込み検索を行う場合でも、基本検索関数の検索条件式に該当する全データ数も同時に自動取得できる機能を設けた。

また、検索時間が設定してある値を超過してしまうなどのエラーが発生することがある。このようなDB処理に関するエラーが発生した状況でも対話そのものは停止させずに対応するために、VoiceXMLの音声認識エラー時のイベントハンドリングと同様に、シナリオの中であらかじめ指定されたエラー用の処理を自動実行できるようにした。

・同一性判定処理

音声対話においては、あるひとつの語の意味(コンセプト)に対して、対応する発話表現(表記)がユーザで多岐にわたる場合がある。例えば、システムが「中華料理店」というコンセプトを持つとき、

表 1: 絞り込み機能の一覧

絞り込み指定子・引数	検索関数による検索結果からの抽出内容
top	検索結果中の最初の1データを取得
tops(件数)	検索結果中の最初の”件数”個のデータを取得
random	検索結果からランダムに1データを取得
randoms(件数)	検索結果からランダムに”件数”個のデータを取得
min(カラム名)	”カラム名”で指定された数値的なデータを持つカラムの中で、最小値を持つデータを取得
mins(カラム名,件数)	”カラム名”で指定された数値的なデータを持つカラムの中で、小さい順に”件数”個のデータを取得
max(カラム名)	”カラム名”で指定された数値的なデータを持つカラムの中で、最大値を持つデータを取得
maxs(カラム名, 件数)	”カラム名”で指定された数値的なデータを持つカラムの中で、大きい順に”件数”個のデータを取得

このコンセプトを指す発話表現が「中華料理屋」「中華」「中国料理」などのようにユーザ間で異なることがある。従来の音声対話システムでは、認識辞書の読み仮名を複数併記するなどしてこの事象に対処していた。

しかし、表現のバリエーションを吸収するには対話サービス毎に語義の扱い方の設定を変えなければならない可能性があり、サービス毎のDBの中で扱えることが望ましい。そこで、ユーザの発話した表現が、システム中のコンセプトと同一か、または異なるかを、上記の基本検索関数を用いることなく調査可能な機能を用意した。この”同一性判定処理”の機能を用いる際には、単語の表記とコンセプトの組み合わせを記した語彙データテーブルをDBとして準備しておけばよい。

この機能により、対話の途中でユーザの使用単語が揺らぐ場合、例えば「中華料理屋で。あ、やっぱり中華じゃなくて…」のような対話でも混乱することなく対応可能である。

5. 他の手法との比較

Polifroni らは対話制御部の汎用化について報告している中で、知識処理やDB処理の汎用化として次の2点について提案している[10]。

- ・ システム応答内容となるDB検索結果のフィルタリング
- ・ ユーザ入力における一般的コンセプトの正規化

まず、DB 検索結果のフィルタリングについては、4.2 節における基本検索関数と絞り込み機能の組み合わせと同様である。ただし、本稿で提案する手法は応答内容の取得のみを目的としておらず、対話処理の中で行われる知識処理全般に RDBMS を利用しようとするものである。さらに、我々の手法には、ユーザに例示などを行う際に、画一的な応答を行わないためにランダムに値を取得する機能があるが、Polifroni らの手法には無い。

なお、Polifroni らは DB から取得した結果の有無や多少に合わせて、応答内容を自動生成する機能を提案している。しかし、システム応答内容はシステムが提供するサービス内容で大きく変化するため、プラットフォームが自動生成する文が適切である場合は少ないと考える。本稿では、応答文の自動生成は行わず、検索結果数に応じた場合分け処理を検索実行時情報の自動取得機能で簡便にしたにとどめた。

Polifroni らのもう一方の提案である、一般的なコンセプトの正規化とは、時間や金額など一般的な入力項目に対する入力発話を、そのバラエティを吸収しながら対話システム内部の意味表現形式に変換することである。Polifroni らは、大量の規則数からなる言語解析ライブラリを準備することでこの枠組みを実現している。我々の同一性判定機能も、ユーザの発話のコンセプトを正規化する性質のものである。しかし、特定のタスクに合わせた規則などは準備していない。対話システム開発者がタスクの単語情報を DB に登録することで本機能が利用可能となる。

6. 今後の課題とまとめ

6.1. 今後の課題

DB と連携する対話システムを構築するには、さらに下記の点が課題として残っている。

- ・依存関係のある複数スロットへの入力の自動処理
自由発話入力技術のように、複数のスロット値を一発話で入力可能にした場合、スロット間に依存関係(例えば路線案内では「路線名」・「駅名」などの関係)のあるスロットへの入力内容が、適切な依存関係を満たしているかを判定する処理が頻出する。これに対しては、本稿において説明した同一性判定と同様の処理と対話制御の自動化または部品化との組み合わせを詳細に検討することにより、開発者の負担が減らせると考えられる。

- ・DB アクセス処理と対話処理の非同期化

本稿で説明した方式では、対話制御の最中に DB アクセス処理が挟まれている。そのため DB アクセスに時間がかかると、その間、利用者には対話システムが停止しているかのように映りかねない。この状況では、その間のバージョンに対応した検索条件の変更が行えないなど、ユーザビリティを低下させる原因ともなる。そこで、DB アクセス処理の最中でもユーザの入力を待ち受ける、または、システムが話題を先に進めることができる等の機能を付加する必要がある。そのために DB アクセス処理の非同期

化を検討する必要がある。

6.2. まとめ

本稿では、実用的な音声対話システムでの知識処理において RDBMS を活用するために、対話システムの開発の観点から問題点を検討した。そして(1)基本検索関数(2)検索結果の絞り込み機能(3)検索実行時情報取得およびエラー処理の自動化(4)同一性判定処理の機能、をもつ DB アクセス仲介モジュールを対話記述言語のインタプリタに追加する構成を提案した。

謝辞 本研究を進めるにあたり、日頃よりご指導いただく NTT サイバースペース研究所音声言語メディア処理プロジェクト 小原永プロジェクトマネージャ、有益な示唆を頂く音声対話インターフェイスグループの皆様にご感謝いたします。また、NTT アイティ社 新原康之さん、井上歩さんはじめ音声コミュニケーション事業部の皆様、NTT アドバンステクノロジ社 石上宏二さんのご協力に感謝いたします。

参考文献

- [1] NTT コミュニケーションズ社 V ポータル.
<http://www.ntt.com/v-portal>
- [2] W3C Voice Browser Activity.
<http://www.w3.org/Voice>
- [3] W3C Multimodal Interaction Activity.
<http://www.w3.org/2002/mmi/>
- [4] 平沢純一, 山本俊一郎, 堀貴明, 大附克年: CTI 向け自由発話対応 音声対話システム RexDialog. 情報処理学会 研究報告 SLP-47, pp.35-40, 2003.
- [5] 甘粕哲郎, 山本俊一郎, 小川厚徳, 篠崎翼, 平沢純一, 山口義一: サイバーアテンダント-自由発話入力に対応したマルチモーダル対話システム. 日本音響学会 春季講演論文集 3-Q-32, 2004-3.
- [6] Kataoka, A., Takahashi, Y., Yamada, S., Asanoma, N. and Furuse, O.: Agent-Mediated Architecture for Efficient Goal-Oriented Communication across Languages. In Proc. of *Asian Symposium on Natural Language Processing to Overcome Language Barriers*, pp.24-30., 2004.
- [7] 野田喜昭, 山口義和, 大附克年, 小川厚徳, 中川聡, 今村明弘: 音声認識エンジン VoiceRex の開発. 日本音響学会 秋季講演論文集 2-1-19, 1999-9.
- [8] 「あすのまち・三鷹」プロジェクト: ビジュアルヘルプデスク 研究プロジェクト報告書.
http://www.asutaka.gr.jp/VHD_result.htm, 2003.
- [9] 中野幹生, 堂坂浩二: 音声対話システムの言語・対話処理. 人工知能学会誌, Vol.17, No.3, pp.271-278, 2002.
- [10] Polifroni, J. and Chung, G.: Promoting Portability in Dialogue Management. In Proc. of *the 7th International Conference on Spoken Language Processing, (ICSLP-2002)*, pp.2721-2724, 2002.
- [11] WWW-データベース連携システム構築法. 日経 BP 社, 1996.
- [12] Inderjeet, S., Beth, S. and Johnson, M.: *Designing Enterprise Applications With the J2EE Platform*. Addison-Wesley, 2002.