

# 日本語音声によるリレーショナルデータベースアクセスの実現

荒川 ゆう子<sup>†,☆</sup> 増 永 良 文<sup>††</sup>

本研究では、日本語音声によるリレーショナルデータベースアクセスを考え、それを実現するシステムの構築を行った。まず、音声による日本語質問文入力から SQL 文への変換、データベースでの質問処理、結果の日本語音声出力という一連の流れを処理できるプロトタイプシステムを市販の音声認識装置、リレーショナルデータベース管理システム、音声合成装置を使い組み上げた。このシステムはそれだけで機能するが、同音語による曖昧性あるいは日本語表現の自然性などに問題があることが分かった。そこでまず第 1 に同音語問題について解決をはかるためにインスタンスキーという新しい概念を定義・導入した。インスタンスキーを用いれば、同音語の入力による曖昧性を解消でき、ユーザの目的とするタプルをインタラクティブに検索することができる。第 2 に、プロトタイプシステムでは日本語の質問文を SQL 文に変換するために必要な選択リスト、表参照リスト、探索条件を、ユーザが発行する日本語質問文にすべて含む必要があるため、日本語質問文が複雑になり、日本語としての自然さに欠けてしまう。この問題を解決するために、ビューと出現語彙表を段階的に使用するという手法を新しく導入・実装した。その結果、結合質問や単純質問を表す日本語文の不自然さを解消することができた。

## Realization of Relational Database Access by Japanese Utterance

YUKO ARAKAWA<sup>†,☆</sup> and YOSHIFUMI MASUNAGA<sup>††</sup>

In order to realize a relational database access interface by Japanese utterance, we developed a prototype system using a voice recognition system, a relational database management system, and a voice synthesis system, which are commercially available. By evaluating this prototype system, we recognized the following two problems to be resolved: The first is the homonym problem due to the fact that there are different words with the same pronunciation which makes the system impossible to identify which word does the pronunciation mean. In order to resolve this problem, we introduced a new key concept named instance key so that the system can retrieve a unique answer by user interaction. The second problem is how to allow natural Japanese speaking, which comes from the restriction that Japanese queries must include all keywords which are necessary to produce equivalent SQL statements. This phenomena is crucial in join-queries. In order to resolve this problem, we introduced a way of using views and an appeared-vocabulary table so that the system can augment necessary information to allow natural but insufficient queries.

### 1. 序 論

#### 1.1 研究の背景と目的

データベースが広く社会に受け入れられて久しいが、リレーショナルデータベースへのアクセスは国際標準データベース言語 SQL に代表されるように、キーボードを用い文字により質問文を書き下し、文字形式

の答えをもらうというのが一般的である。しかしながら、データベースアクセスのためにキーボードはあるがキーボードの使えない状況、たとえば手がふさがっている場合（執筆中、操作盤を操作中、あるいはすでに他の目的でキーボードを操作中など）や、あるいは手はふさがっていてもキーボードを操作できない場合（コンピュータに不慣れな初心者、コンピュータを使えない老人や盲人がユーザである場合など）、さらにキーボードは使える状況ではあるが SQL を知らなかったり、SQL を知っていても緊急にデータベースにアクセスしたい場合、加えてキーボードのない状況（たとえばコンピュータを携帯していない外出先など）では、文字以外のメディア、特に人間にとって最も自然な意志伝達手段である音声によるデータベースアク

† 図書館情報大学図書館情報学研究科  
Graduate School of Library and Information Science,  
University of Library and Information Science

☆ 現在、富士通株式会社  
Presently with FUJITSU LIMITED

†† 図書館情報大学図書館情報学部  
Department of Library and Information Science, Uni-  
versity of Library and Information Science

セスの実現が期待される。

このような音声によるデータベースアクセスを実現するために、本研究では音声認識分野における音声インタフェースの研究とデータベース分野における自然言語インタフェースの研究の融合をはかり、音声インタフェースによるデータベースアクセスの実現を試みた。我々は現在利用可能な音声認識装置や音声合成装置を使用して、日本語音声によるリレーショナルデータベースアクセスができるプロトタイプシステムの構築を行い、研究を遂行してきた<sup>1),2)</sup>。構築したプロトタイプシステムでは音声で入力された自然言語による日本語質問文は SQL 文に変換され、リレーショナルデータベースへの質問文となり、結果リレーションは日本語音声で出力されるという一連の流れが完成している。本稿では、まずこのプロトタイプシステムの概要について述べ、さらに、このプロトタイプシステムを評価した結果明らかになった、リレーショナルデータベースへ日本語音声でアクセスする際に生じる基本的な2つの問題、すなわち日本語発話における同音語から引き起こされる曖昧性の問題と質問発行時の日本語発話における自然さの達成についての問題の解決策を述べる。また、このシステムの実用化に向けて、使用した音声認識装置の認識率の評価も行った。以下本論文では関連研究に言及した後、2章で日本語音声によるデータベースアクセスのプロトタイプシステムの概要とその問題点について述べる。3章ではプロトタイプシステムにおける同音語問題の解決について述べ、4章ではより自然な日本語発話に対する問題の解決について述べ、プロトタイプシステムの改良をはかる。5章では改良されたプロトタイプシステムの評価を行う。6章は結論である。

## 1.2 関連研究

本研究に関連する研究としては、データベースの自然言語インタフェースに関する研究とコンピュータの音声インタフェースに関する研究の2種類に大別される。まず、自然言語でデータベースにアクセスするシステムはすでにいくつか研究が行われているが、自然言語文の認識法の違いにより、大別すると2つアプローチがある。1つは入力された自然言語文を構文解析や語彙解析、意味解析といった自然言語処理を行って SQL などの検索言語を生成する方法である<sup>3)~8)</sup>。この中で有用性が高いと思われるのが「IF-Kit」<sup>7)</sup>である。IF-Kit は入力された日本語質問文（漢字かな交じり文）から SQL 文を生成してリレーショナルデータベースにアクセスできるシステムである。また、一般に自然言語処理を用いれば柔軟な表現が入力可能とな

るが、その柔軟性を広げようとする解析処理が複雑かつ膨大になってしまい、実用性を考えると制限された表現になってしまう。そのため、構文・意味解析を高速に、少ない作業領域で行えるツールも開発されてきた<sup>9)</sup>。さて、もう1つのアプローチは、本システムで採用した方法であるが、構文解析や意味解析といった自然言語処理を用いずに、音声認識の一手法であるシンタックス記述方式を利用してパターンマッチングのみで自然言語文の認識を行い、必要な単語の抽出を行って認識した自然言語文に等価な SQL に変換する方法である。この手法では入力可能な質問のパターンをあらかじめ設定しておく必要があるため、質問の表現は制限されたものとなるが、質問文の解析を行う必要がない。またアプリケーションごとにデータベースアクセスの範囲をある程度設定すれば、このように制限された質問文でも十分実用性があると考えられる。このような理由から本研究ではこの方法を採用している。

次に、より一般的なコンピュータの音声インタフェースに関する研究で本研究と関連があるのは、その音声認識技術をヒューマンインタフェースとして応用している研究である。その中で音声によるデータベース検索を行っているものとして、「たびの電子ガイドシステム：TARSAN」<sup>10)</sup>、「富士山観光案内」をタスクとする音声対話システム<sup>11)</sup>、「SIMPLA」<sup>12)</sup>があげられる。まず TARSAN は CD-ROM に収録された観光情報を全文検索することができ、文献 11) のシステムは知識データベースに格納された富士山観光案内情報を検索することができる。両者とも音声対話システムの応用であるため音声認識部とユーザとの応答部に力を入れたシステムとなっており、データベース検索は SQL などの検索言語を用いてはいない。その点で我々の研究に一番近いシステムは SIMPLA である。SIMPLA はリレーショナルデータベースに対するマルチモーダルな自然言語インタフェースであり、質問を日本語音声で入力できるが、質問を単語ごとに区切って言う必要があり、たとえば文節単位で発話できるように改良することが課題として残っている。検索結果は、表情報はテキストとしてディスプレイに表示され、検索件数などは音声で出力される。つまり SIMPLA はユーザがコンピュータの前に座っていることを前提としたシステムである。しかし我々のシステムでは質問は文章で、連続音声で発話することができる。また、応用分野として電話を用いた遠隔地からのデータベースアクセスや仮想空間でのデータベースアクセスを考慮に入れているため、音声のみのインタフェースを実

装しており、コンピュータの前にいる必要がない点でSIMPLAとは異なる。

2. プロトタイプシステム

2.1 想定アプリケーション

本研究では音声によるデータベースアクセスが有効であるような場面を想定することからデータベース作成作業を開始した。それはある商品販売会社のセールスマンが携帯電話機を所持しながら、全国津々浦々を訪問販売しており、セールスマンは出先で客と対応しているときに、電話で商品に関する質問や在庫に関する質問などをその場で質問でき答えを聞ける状況とした。そのために構築したサンプルデータベースは図1の3つのリレーションから成っている。

2.2 プロトタイプシステム

2.2.1 システム構成

日本語音声をSQL文に変換してリレーショナルデータベースへ質問を発行するためには、次の処理を行う必要がある。

- (1) 日本語音声を日本語文に変換
- (2) 日本語文をSQL文に変換

(1)の過程は音声認識技術の発達にともない、様々な方式が開発されてきた。後述するシンタックス記述方式もその1つである。この方式では認識した文をセンテンスパターンに従った文字列で出力するため、これをSQL文に変換する(2)の処理が必要である。質問処理の結果は結果リレーションとして返されるため、その結果リレーションをまたユーザへ出力する際に自然な日本語音声に変換する必要がある。さらに、本研究で使用している音声認識装置は、認識に失敗すると入力した質問文とは違ったセンテンスを返したり、ま

たセンテンスパターンとして登録されていない文が入力されても、登録されているいずれかの文を出力してしまうことがある。するとそのセンテンスがSQL文に変換され、質問処理が行われてしまい、ユーザが意図する答えとは違ったものが返される恐れがある。また、音声認識装置がユーザの入力をいずれのセンテンスパターンにもマッチしないと判断した場合には、何の応答もせず次の入力を待つ状態になってしまう。このような場合を考慮にいれ、ユーザの入力が正しく行われたかを確認するモジュールも必要になる。このモジュールでは入力された日本語質問文をユーザに確認してもらうための確認文を生成するモジュールも必要となる。

以上の点を考慮にいれて、本研究では以下のようにプロトタイプシステムを構築した。このシステムでの処理の流れを図2に示す。まず、アクセスの対象となるデータベース(=リレーション群:図1)を構築し、ある特定の利用環境下でのデータベースアクセスを考え、その場合での質問文(群)を想定し、その質問文に対応するセンテンスパターンを設定した。さらに、上記で示した変換モジュールを次のように開発した(図2)。

- (a) センテンス確定モジュール: 音声認識装置によって認識されたセンテンスのセンテンスパターンを同定するモジュール。
- (b) SQL生成モジュール: 認識されたセンテンスを上記モジュールが同定したそのセンテンスパターンを使って、SQL文に変換するモジュール。
- (c) SQL処理モジュール: 上記モジュールで生成されたSQL文でリレーショナルデータベースへ質問を発行し、結果として結果リレーションを受け取るモジュール。
- (d) 質問結果発声モジュール: 結果リレーションを音声合成装置の入力系列に変換するモジュール。
- (e) 入力確認モジュール: ユーザの入力文が正しく入力・認識されたかを確認するためのモジュール。入力が確認された場合は、その入力が正しく認識

商品番号	商品名	型	メーカー	定価
001	Aptiva_560	デスクトップ	IBM	280
002	Aptiva_770	モニタワ	IBM	300
005	PC-9821Cb10	デスクトップ	NEC	298
032	Libretto20	ノート	東芝	178
...	...	...	...	...

商品番号	在庫量
001	100
002	300
005	200
032	250
...	...

社員番号	社員姓	社員名	読み姓	読み名	所属	給与	性別	内線番号	入社年
880271	小島	晃	こじま	あきら	営業	530000	M	2738	1988
880277	小島	早苗	こじま	さなえ	総務	500000	F	5678	1988
900195	見島	直樹	こじま	なおき	開発	400000	M	1234	1990
800791	山田	洋一	やまだ	よういち	営業	900000	M	2738	1980
...	...	...	...	...	...	...	...	...	...

図1 サンプルリレーショナルデータベース  
Fig.1 A sample relational database.

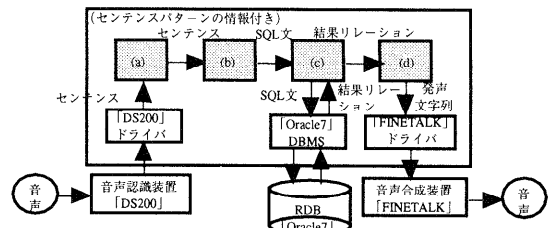


図2 プロトタイプシステムの構成  
Fig.2 Structure of the prototype system.

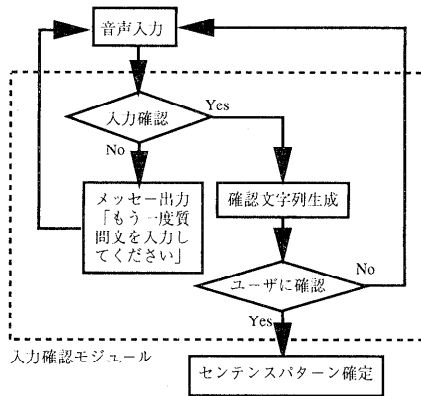


図3 入力確認モジュール

Fig. 3 Input confirmation module.

されたかをユーザに問い合わせるための確認文を生成する(図3)。図2には現れていないがユーザとのインタラクションの際に使用される。

プロトタイプシステムでは(株)オーヂス総研の音声認識装置「DS200」、日本オラクル(株)のリレーショナルデータベース管理システム「Oracle7」、(株)明電舎の音声合成装置「FineTalk」、ワークステーションにはSun SPARCstation2を使用した。

2.2.2 音声認識装置「DS200」

本研究で使用している音声認識装置「DS200」は、セットアップの際に男性/女性、英語/日本語用をセットするだけで、不特定話者の連続的に発声された音声認識するシステムである。音声認識にはシンタックス記述方式を採用している。シンタックス記述方式とは音声認識をセンテンスパターンを用いて行う方式で、この方式は米国 Speech Systems 社製の音声認識装置「DS200」に実装されており、(株)オーヂス総研により日本語に対応するように基本アルゴリズムの改訂が行われた<sup>13)</sup>。センテンスパターンとは、音声認識装置において音韻デコーダが入力された音声をデコード(音韻コード列を文字列に変換する処理)する場合に、どのセンテンスが発話されたかを決定するためのルールであり、シンタックスソースファイルにあらかじめ記述しておく。このセンテンスパターンを用いれば、SQL文の生成を自然言語処理を行うことなく、パターンマッチングで行うことができる。

本方式では質問文の構文をあらかじめいくつか決めていたので、限定された構文しか認識できないばかりか、その構文に登録された単語しか認識できないという問題がある。つまり未登録語をシステムが認識できないので、データベース更新時に未登録語をユーザ

```

SS -> 探索条件_ $[の] リレーション名_ *
[の] 検索属性_ は [わ]
探索条件_ -> 商品名 [しょうひんめい] % [か] 商品名_
リレーション名_ == 商品 [しょうひん]
検索属性_ == 定価 [ていか] メーカー [めーかー]
商品名_ == Aptiva_560 [あぶていばごうろくまる]
    
```

図4 単純質問のセンテンスパターン例

Fig. 4 A sentence pattern for simple queries.

が発声してもシステムは認識できない。したがって、データベースの更新には制限が出る方式である(このことはデータベース検索の立場からは特段の問題は起こらない)。

ここで、本研究で使用するセンテンスパターンの作成法について述べる。本研究ではセンテンスパターンをひな形として日本語質問文を作成し、その日本語質問文からSQL文を生成するため、その質問を表すSQL文のSELECT句の選択リスト、FROM句の表参照リスト、WHERE句の探索条件をすべてユーザが発話した日本語質問文から抜き出したい。このことを考慮して単純質問についてのセンテンスパターン作成法を論じる。

単純質問は1つのリレーションに対して発行される質問である。したがって、あるリレーションの属性値(の集合)を指定して、そのリレーションの別の属性値(の集合)を求めようとする質問である。たとえば、リレーション商品を対象にして「商品名がAptiva\_560の商品の定価は」という質問が典型的と考えられるが、図4に示すセンテンスパターンを音声認識装置に設定しておく、これは「商品名% Aptiva560 \$商品\*定価は」というセンテンスとして認識される。したがって、選択リストは“定価”、表参照リストは“商品”、そして探索条件は“商品名=‘Aptiva\_560’”であることがパターンマッチで分かる。さらに、上述の議論から単純質問は一般に図4に示されるセンテンスパターンで記述できる。

2.2.3 SQL文生成までの処理

1つのセンテンスパターンは、ユーザが発行するある質問パターン(本論文ではより具体的にSQLパターンと呼ぶ)に対応している。したがって、音声認識装置によって認識されたセンテンスがどのセンテンスパターンに一致したのかが分かればセンテンスの意味する日本語質問文に対応するSQL文を生成することができると思われる。

まず認識されたセンテンスがどのセンテンスパターンにマッチしているのかを割り出す処理を行う。この処理が必要なのは音声認識装置「DS200」が発声された日本語質問文の認識をセンテンスパターンを用い

日本語質問文：商品名が Aptiva\_560 の商品の定価は  
 センテンスパターン：図 4 参照

SQL パターン：	SELECT	検索属性
	FROM	リレーション名
	WHERE	商品名='商品名';
質問の SQL 文：	SELECT	定価
	FROM	商品
	WHERE	商品名='Aptiva_560';

図 5 日本語質問文、センテンスパターン、SQL パターン、SQL 文の対応

Fig. 5 Correspondence among Japanese-query, sentence pattern, SQL-pattern and SQL statement.

て行い、その結果としてセンテンスを出力してくれるものの、数あるセンテンスパターンのうちのパターンに合致したのかの情報は出力してくれないからである。したがって、我々は出力されたセンテンスのパターン認識によりセンテンスパターンの割り出しを行った。これを行うのがセンテンス確定モジュールである（図 2(a)）。

次に、センテンスパターンと SQL パターンの関係付けを簡単な例題で見てみる。ここで SQL パターンは SQL 文の SELECT 句の選択リスト、FROM 句の表参照リスト、WHERE 句の探索条件の値がそれぞれ変数として定義される表現をいうと定義する。図 5 にセンテンス、センテンスパターン、対応する SQL パターン、SQL 文の一例を「商品名が Aptiva\_560 の商品の定価は」という単純質問に対して与えてみる。センテンス確定モジュールにより確定された文字列は図 5 のような流れで SQL 文に変換される（図 2(b)）。

#### 2.2.4 動的 SQL 文によるデータベースアクセスと結果処理

前項の SQL 変換により、日本語音声による質問文を SQL 文に変換することが可能となった。次に、作成された SQL 文でリレーショナルデータベースへアクセスする方法を説明する。

埋込み型 SQL 文などの静的 SQL 文では、実行する時点でその SQL 文の構成がすべて分かっている必要はない。しかし、実行する SQL 文の構成が実行時まで分からないものや、実行のたびに異なるものもある。このようなものは、各種の SQL 文を実行時に受け入れて処理しなければならない。このように実行時に SQL 文を作り出して処理を行う形式が動的 SQL 文によるプログラミングであるが、本研究における質問処理は対話式に行われるため、センテンスパターンは設定されているものの、その個々の変数にどの単語が入力されるのかは実際に質問が発せられてからでないと分からない。つまり本研究では、質問が発せられ

るたびにその質問文を SQL 文に変換するため、SQL 文の構成は実行時まで分からない。よって動的 SQL 文を使用してリレーショナルデータベースへアクセスする必要がある。

動的 SQL 文を使用することにより、リレーショナルデータベース管理システム「Oracle7」で質問処理を行い、検索結果を結果リレーションとして出力することが可能となった（図 2(c)）。動的 SQL 文を使用するアプリケーションプログラム内では、検索結果は変数に入れられるためその結果を答えのセンテンスとして適当なものに変換し（図 2(d)）、音声合成装置「FineTalk」を用いて音声で出力させた。

以上により、音声による日本語質問文入力から SQL 文への変換、リレーショナルデータベースでの質問処理、結果の日本語音声出力という一連の流れを処理できるプロトタイプシステムが完成した。このプロトタイプシステムを評価した結果、次のような問題点が明らかになった。

- (1) 同音語の問題
- (2) 日本語発話における自然性の問題

次章からこれらの問題点の解決法について述べる。

### 3. 同音語問題とその解決法

#### 3.1 問題点の同定

音声は人間にとって最も自然な情報伝達手段の 1 つである。よって、音声によるインタフェースはシステムをより使いやすくするために必須のものであると考えられるが、その音声入力にも問題点がある。それは同音語の識別である。たとえば「児島」、「小島」、「古島」は文字は異なるが発声すればみな「こじま」であり、発話からではそれらを区別できない。これが同音語の問題である。さらにこの問題を入力時と出力時に分けて述べる。

##### (a) 入力時の問題

本研究で用いた音声認識装置は、センテンスパターンの登録単語として「児島[こじま]」、「小島[こじま]」、「古島[こじま]」をこの順に設定した場合、「こじま」と音声で入力すると、最初に登録されている「児島」を返してしまう。したがって、機械的制約により小島と古島はそもそも認識できない。そこで、人名など同音語問題を引き起こす可能性のある属性については、(i) センテンスパターンの登録単語を「読み」（ひらがな）で設定し、リレーションに「読み」という新たな属性を加えることで読みで検索を行い検索漏れをなくし、さらに、(ii) その中から所望のタプルを一意に

識別するための特別の手立てを導入する，ことにより問題解決をはかる必要がある。

#### (b) 出力時の問題

音声出力では漢字の情報までは伝えられないため，出力された結果に対して的確な判断ができない場合が生ずる。このような場合は名前だけでなく他の情報も付け加えて出力するなどの機能が必要となる。たとえば「えいぎょうのこじま」は「そうむのこじま」と異なる「こじま」を伝達できる。

このように，音声をういたシステムでは同音語の問題は様々な局面で問題を引き起こすことが考えられる。これは単に音声認識の問題にとどまらず，同音語を前提にデータベースアクセスをどうするかという本質的な問題としてとらえられるべきである。このような認識から本研究ではリレーショナルデータベースにおける候補キー（以下キー）の問題を再考し，特に(a)の日本語音声による質問の入力時に発生する同音語による曖昧性を解消する目的で，インスタンスキーなる概念を新規に定義・導入しこの問題に実用的な解決を図った<sup>14)</sup>。次節からインスタンスキーについて詳しく述べる。

### 3.2 インスタンスキーの定義

ある時点のリレーションにおいてキーとしての役割（すなわちタプルの一意識別能力）を担う属性（の組）をインスタンスキーというとする。インスタンスキーの定義を次に示す。

#### 【定義1】インスタンスキー

$R(A_1, A_2, \dots, A_n)$  をリレーションスキーマとし，時刻  $\tau$  における  $R$  のインスタンスを  $R_\tau$  とする。  $K = \{A_{k1}, A_{k2}, \dots, A_{kp}\} (1 \leq k1 < k2 < \dots < kp \leq n)$  を属性の集合とすると，  $K_\tau$  が  $R_\tau$  のインスタンスキーと呼ばれるのは，次の条件を満たすときである：

- (i)  $(\forall t, t' \in R_\tau)(t[K_\tau] = t'[K_\tau] \Rightarrow t = t')$
- (ii)  $(\forall H \subset K_\tau)(\neg((\forall t, t' \in R_\tau)(t[H] = t'[H] \Rightarrow t = t')))$

つまり，インスタンスキーは指定された時刻でのみタプルの一意識別能力を持つ属性の極小の組をいう。一般にインスタンスキーは複数あるかもしれない。また，ビューにおいてもインスタンスキーは実表同様に定義できる。なお，リレーションスキーマ上の関数従属性（functional dependency, FD）を用いて定義・計算される従来のキーを以後FDキー（functional dependency key）と呼ぶことにする。FDキー  $K$  は  $(\forall \tau)(\forall t, t' \in R_\tau)(t[K] = t'[K] \Rightarrow t = t')$  を満たす極小集合と定義される。

### 3.3 FDキーとインスタンスキーの関係

ここでFDキーとインスタンスキーの関係について述べる。確かにFDキーはタプルの唯一識別能力を持つが，FDキーはリレーションスキーマを設計した時点で，そのスキーマごとに定義されるため，インスタンスがその後どのように変化しようとも永久的にキーであり続ける。またこのために，キー制約に抵触するタプルの挿入，削除，修正は受け付けられない。しかしインスタンスキーの場合はタプルの挿入，削除，修正によってそれがインスタンスキーでなくなるようなことがあっても一向にかまわない。インスタンスキーはたとえばリレーション社員の属性{給与}の値がすべて違っていたらその属性{給与}はインスタンスキーになるように，どの属性の組もインスタンスキーになる可能性を持っている。その意味で，インスタンスキーを用いる方がタプルの一意識別の可能性はFDキーを用いたときより高くなる。さらにどの属性がインスタンスキーとなるかはFDキーとは一般的には関係がない。FDキーの真部分集合がインスタンスキーとなることもある。しかしFDキーはタプルの唯一識別子というその性質より，最悪の場合インスタンスキーになるという意味で次の命題が成立する。

【命題1】FDキーの部分集合（真部分集合である必要はない）の中に必ずインスタンスキーとなる属性の組が存在する。

【証明】  $R(A_1, A_2, \dots, A_n)$  をリレーションスキーマとし，時刻  $\tau$  における  $R$  のインスタンスを  $R_\tau$  とする。  $K$  を  $R$  のFDキーとし，  $H$  を  $K$  の真部分集合とする。時刻  $\tau$  においてたまたま  $(\forall t, t' \in R_\tau)(t[H] = t'[H] \Rightarrow t = t')$  が成り立てば，定義1より  $H$  はインスタンスキーである。もしこのような  $H$  が存在しなくても  $K$  は  $R$  のFDキーなので  $(\forall \tau)(\forall t, t' \in R_\tau)(t[K] = t'[K] \Rightarrow t = t')$  が成り立ち定義1より  $K$  はインスタンスキーとなる。 Q.E.D.

次節でインスタンスキーの算出方式について述べる。

### 3.4 インスタンスキーの算出アルゴリズム

以下で時刻  $\tau$  におけるインスタンスキーの算出アルゴリズムを説明する。まず，対象となるリレーションの全属性リストからユーザが発行した質問文のSQL文（図6(a)）のSELECT句に現れた属性リストとそのWHERE句に現れた属性リストを差し引いたビューを生成し，“CHUKAN”と名付ける（図6(b)）。次に，生成された中間結果リレーションの最適選択能力とそれぞれの属性に対する選択係数（selectivity factor）<sup>15)</sup>を求める。ここで最適選択能力  $G$  は，  $G = (1/\text{中間結果リレーションのタプル数})$  で表され，この値と

```

(a) 質問文のSQL文
select X
from R
where A = 'a';

(b) "CHUKAN" ビュー生成のためのSQL文
create view CHUKAN as
select * - (X, A)
from R
where A = 'a';

```

図6 “CHUKAN” ビュー生成のためのSQL文  
Fig. 6 SQL statement for generating view “CHUKAN”.

理を行うフローチャートを図7に示す。

3.5 インスタンスキーの評価と問題点

インスタンスキーを導入することで同音語の曖昧性が原因で複数となってしまった結果から、ユーザの意図するタプルを一意に識別することが可能となった。またインスタンスキーは同音語による曖昧性を解消するだけではなく、一般的に複数になってしまった結果から所望のタプルを一意に識別することができる。しかし、この処理を評価した結果次のような問題点が明らかになった。

(1) タプルが同定できなかった場合

これは、ユーザがシステムに問い掛けられたどのインスタンスキーの値をも知らなかった場合に起こりうる。この場合はタプルが一意に識別されないわけだが、これはユーザ自身に所望の結果に対する識別能力がなかったためであって、システム側ではどうすることもできない。そのような場合にはユーザに複数の結果をそのまま出力するのか、出力は行わないのか。あるいは、初めの何件かだけ出力するのかなど、出力形式を選択してもらい問題解決にあたるのが考えられる。本研究においてはシステムからの問合せに対してユーザに出力件数を音声で入力してもらい、その件数だけ出力している。ユーザが意図した結果がそこになければさらに出力を要求できるので、特段の問題は発生しなかった。

(2) 計算量の問題

3.4節のアルゴリズムに従ってインスタンスキーを探索する場合の計算量は中間結果リレーシンの属性数を  $n$  とすると、そのすべての組合せについてインスタンスキーかどうかの判定を行うため、単純に計算すると最初のインスタンスキーを導出するのに、最悪の場合  $2^n - 1$  回の計算を必要とし、指数時間のオーダーとなってしまふ。しかし1MIPSの計算機であれば  $n$  の値が20以下の場合には1秒以下で処理できる(図8上段)<sup>16)</sup>。本研究で使用している計算機(Sun SPARCstation 2)は28.5MIPSの処理能力を持つため<sup>17)</sup>、計算時間はさらに早くなり(図8下段)、また属性数が20以上のリレーシオンは、実装ではあまりないと楽観的にとらえ、計算の複雑さの問題は、理論上は問題であるが実際には問題にならないと結論した。

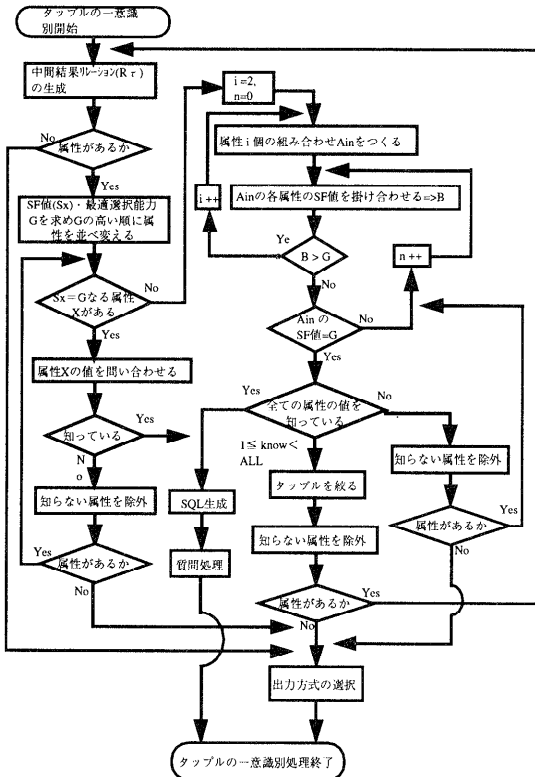


図7 インスタンスキーによるタプルの一意識別処理  
Fig. 7 Process of unique identification of a tuple using instance keys.

等しい選択係数を持つ属性の極小の組をインスタンスキーとする。選択係数は属性の選択識別能力の高さを示す値で、(1/属性値の種類数)で表される。たとえば、リレーシオン社員のある時点におけるインスタンスが4本のタプルから成り、それらの所属値がそれぞれ営業、開発、開発、総務であった場合、属性所属の選択係数は1/3となる。このようにして算出されたインスタンスキーの値をユーザに問い合わせることでタプルを一意に識別する。ユーザがインスタンスキーの値を知らなかった場合には新たに別の中間結果リレーシオンを作成し、選択係数を計算して新たなインスタンスキーを探索する。

インスタンスキーを算出してタプルの一意識別処

Time complexity function	Size n					
	10	20	30	40	50	60
$n$	.0001 second	.0002 second	.0003 second	.0004 second	.0005 second	.0006 second
$n^2$	.0000035 second	.0000070 second	.0000105 second	.0000140 second	.0000175 second	.0000211 second
$n^3$	.001 second	.004 second	.009 second	.016 second	.025 second	.036 second
$n^4$	.00004 second	.00014 second	.00032 second	.00056 second	.00088 second	.00126 second
$n^5$	.001 second	.008 second	.027 second	.064 second	.125 second	.216 second
$n^6$	.003509 second	.112 seconds	.852 seconds	3.59 seconds	10.96 seconds	27.28 seconds
$2^n$	.001 second	1.0 second	17.9 minutes	12.7 days	35.7 years	366 centuries
$3^n$	.059 second	58 minutes	6.5 years	3855 centuries	$2 \times 10^8$ centuries	$1.3 \times 10^{13}$ centuries
	.002072 second	2.03 minutes	83.6 days	135 centuries	$8 \times 10^6$ centuries	$4.7 \times 10^{11}$ centuries

\*表中の点線は上段は1MIPSの下段は28.5MIPSの処理速度を持つ計算機の場合を表す。

図8 指数関数と多項式関数の処理時間の比較 (1 MIPS と 28.5 MIPS の場合)<sup>16)</sup>

Fig. 8 Comparison of several polynomial and exponentialtime complexity functions (case of 1 MIPS and 28.5 MIPS).

## 4. より自然な日本語音声の受理

### 4.1 日本語質問文の不自然さとは

本研究ではセンテンスパターンに基づいて質問文を設定している。また、そのセンテンスパターンからパターンマッチングでSQL文を生成するので、センテンスパターンには少なくともSQL文のSELECT句、FROM句、WHERE句に対応する選択リスト、表参照リスト、探索条件が組み込まれていなければならない。このことは「自然な日本語による質問」の妨げになる。

たとえば、単純質問の典型例であった「商品名がAptiva\_560の商品の定価は」は、この質問を表すSQL文の選択リスト、表参照リスト、探索条件に具体的な値を指定するために、「定価」、「商品」、「商品名がAptiva\_560」という語句を含んでいる(図5参照)が、「Aptiva\_560の定価は」というより自然な質問文でも同じ質問意図を表現することができる。さらに、複数のリレーションを対象とした結合(join)演算を含むような結合質問の場合、SQL文のWHERE句に結合条件を指定しなければならないため、日本語質問文にもその複数のリレーション名を含めなければならない。典型例として「商品名がAptiva\_560の商品の在庫の在

庫量は」という質問文があげられるが、これは図1(a),(b)に示したりレーション商品と在庫を使用して商品名がAptiva\_560である商品の在庫量を求めたい場合の質問文である。これは前述の単純質問に比べると、結合に必要なリレーション名“在庫”を1つ余計に指定しなければならないため、その分自然さを欠く表現となっている。理想的には、この質問は「Aptiva\_560の在庫量は」でシステムが受け付けてくれるとありがたい。

我々はこのようなより自然な日本語質問文を受理させるために、以下のような段階的解決を図った。

**第1段階** ビューを用いることによって、結合質問を表す日本語質問文の不自然さを単純質問レベルの日本語質問文のそれに減ずる。

**第2段階** 単純質問を表す日本語質問文の不自然さは出現語彙表という新しい概念を定義・導入することで、解消する。

次節からこれらの2つの解決法を説明する。

### 4.2 結合ビューの導入による不自然さの解消 (第1段階)

結合質問の日本語文の複雑さを単純質問のレベルに還元するために、質問の対象となると想定されるリレーションの結合(join)結果を結合ビューとしてあらかじめデータベース内に定義することを考える。ビューは1枚の仮想的リレーションとなるのでそのビューに質問をかけるようにすれば、結合質問の日本語文は単純質問のセンテンスパターンにより規定される。たとえば、図1(a),(b)のリレーション商品、在庫に対しては結合演算を施した結果として図9のようなビュー商品\_在庫を用意しておけば、先程の「商品名がAptiva\_560の商品の在庫の在庫量は」という質問文は「商品名がAptiva\_560の商品\_在庫の在庫量は」という単純質問のセンテンスパターンで問合せができ、そのSQL文は図10ようになる。

この方法では、結合ビューがデータベース内にあらかじめ定義されていればユーザはそれを指定して(単純質問レベルの)質問がかけられるが、ユーザの意図する結合ビューが定義されていなかった場合には、不自然な日本語である結合質問のセンテンスパターンで質問しなければならないというデメリットがある。なお、質問文中にビュー名である「商品\_在庫」を指定するのは不自然であると感じ、単に「商品名がAptiva\_560の在庫量は」と質問できないのかと考えるかもしれないが、その解決策は次節で示す。



ビュー 商品\_在庫

商品名	在庫量
Aptiva_560	100
Aptiva_770	300
PC-9821Cb10	200
Libretto20	250
...	...

図 9 ビュー-商品\_在庫  
Fig. 9 View Product\_Stock.

```

SQL文
SELECT 在庫量
FROM 商品_在庫
WHERE 商品名 = 'Aptiva_560';
    
```

図 10 商品\_在庫上の SQL 文  
Fig. 10 SQL statement on Product\_Stock.

4.3 出現語彙表の導入による不自然さの解消 (第 2 段階)

前節で述べたようにビューを用いることによって結合質問を表す日本語文の不自然さを単純質問のそのレベルに還元できた。本節では、単純質問を表す日本語質問文の不自然さを出現語彙表という新しい概念を定義・導入することで、解消できること示す。これは「Aptiva.560の定価は」というような簡素化された質問文には対応する SQL 文生成に必要な情報がすべて含まれていないので、不足している情報を出現語彙表を用いることによって補完しようとするものである。

出現語彙表とは、データベースに格納されている全リレーションに属性値として出現している値をその属性名とリレーション名(ビュー名も含む)とともに保持しておく表である(図 11)。

出現語彙表を用いて SQL 文を生成する方法を次に示す。まず、日本語音声による質問文入力から質問結果の日本語音声出力までの処理の流れを図 12 に示す。このうち点線で囲った部分が出現語彙表を用いた処理である。この処理を行うには、次の準備が必要である。

- (1) 自然な日本語質問文のセンテンスパターンを設定する。
- (2) 上記センテンスパターンと 1 対 1 に対応する SQL パターンを設定する。

センテンスパターンの種類は、その日本語質問文で表される探索条件の情報量により以下の 2 種類に分けられる(必然的に SQL パターンも 2 種類になる)。

(a) 「探索条件\_ \* [の] 属性名\_ は」(探索条件\_については図 4 参照)

出現語彙表

属性値	属性名	リレーション名
Aptiva_560	商品名	商品
Aptiva_770	商品名	商品
200	在庫量	在庫
380	定価	商品
こじま	読み姓	社員
...	...	...

図 11 出現語彙表  
Fig. 11 Appeared-vocabulary table.

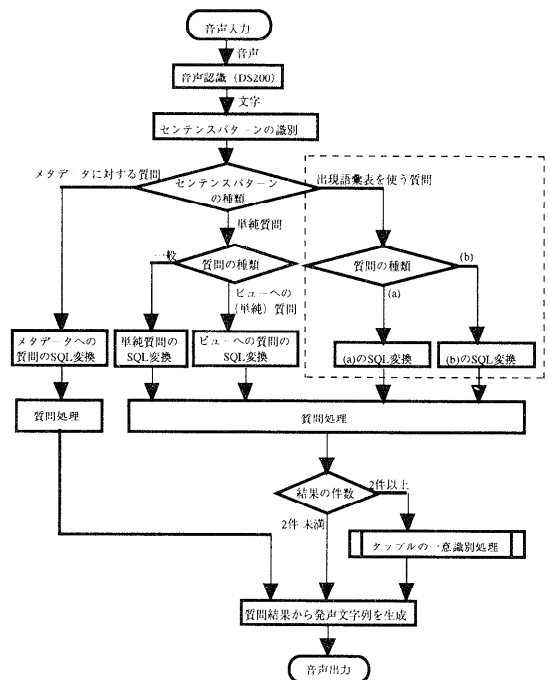


図 12 日本語音声によるデータベースアクセスの処理の流れ  
Fig. 12 Process flow of the prototype system.

(b) 「属性値\_ \* [の] 属性名\_ は」

これらのセンテンスパターンに対する SQL パターンは図 13 (a), (b) となる。(a), (b) 両方で必要なのがリレーション名についての情報である。(b) ではさらに探索条件中の属性名についての情報も必要である。これらの情報を出現語彙表から抽出する。これらのセンテンスパターンの典型例としては (a) 「商品名が Aptiva.560 の定価は」、(b) 「Aptiva.560 の定価は」があげられる。

次に、出現語彙表を用いて SQL 文を生成する方法を説明する。単純質問への SQL 変換では、入力された日本語質問文から SQL パターンのそれぞれの変数

	(a)		(b)
SELECT	属性名 <sub>1</sub>	SELECT	属性名 <sub>1</sub>
FROM	R	FROM	R
WHERE	探索条件;	WHERE	A = '属性値 <sub>1</sub> ';

図 13 出現語彙表を用いる場合の SQL パターン  
Fig. 13 SQL-pattern for appeared-vocabulary table.

に対応する単語を抽出するだけでよかったが、出現語彙表を使用するこの方法では、それに加えて SQL パターンに記載されていない情報を出現語彙表から選択しなければならない。そのためには次の処理が必要である。

#### (1) リレーション名 R の決定

1. 出現語彙表から、属性値が“属性値<sub>1</sub>”であるリレーション名の集合  $S_1$  を選択
2. 出現語彙表から、属性名が“属性名<sub>1</sub>”であるリレーション名の集合  $S_2$  を選択
3. 集合  $S_1$  と  $S_2$  の共通集合 T をとる
4. T の要素数が 1 つならそれが R となり、複数の場合は次節の方法を用いて R を決定する。

#### (2) 属性名 A の決定

属性値が“属性値<sub>1</sub>”で、かつリレーション名が、(1) で決定されたリレーション名 R である属性名 A を出現語彙表から選択あとはそれぞれの SQL パターンに上記で決定した R, A を挿入することで、SQL 文を完成できる。

このように従来のセンテンスパターン、SQL パターンを使用した SQL 変換方式に出現語彙表を用いた場合の処理を加えることで、単純質問の場合でもより自然な日本語質問文の SQL 文を生成することが可能となった。なお出現語彙表のサイズ P は、リレショナルデータベースを  $DB = \{R_1, R_2, \dots, R_n\}$  とし、 $ct(R_i)$  と  $ca(R_i)$  を各リレーション  $R_i$  の総タプル数と総属性数とするときに、 $P \leq ct(R_1) \times ca(R_1) + ct(R_2) \times ca(R_2) + \dots + ct(R_n) \times ca(R_n)$  となる。一方、リレーションに更新があった場合に出現語彙表をどのようなタイミングで更新してゆくかについては、基本的に即時更新 (immediate update)、遅延更新 (deferred update)、スナップショット (snapshot) の 3 種が考えられる。どの方法をとるかはアプリケーションによるが、本研究ではスナップショット法を採用している。スナップショット法は維持が容易というメリットがあるが、実リレーションの更新がまだ出現語彙表に反映されていないときに出現語彙表へのアクセスがあると、不整合が起こる場合があるというデメリットがある。

#### 4.4 出現語彙表導入により発生する問題の解決

出現語彙表は (単純質問レベルの) 自然な日本語質問文から等価な SQL 文を生成するために必要な情報を補うために使われる。しかしながら、一般には同じ属性名を持っている異なるリレーションが存在したり、同じ属性値をとっている異なる属性が 1 つのリレーションに複数存在することがあるために、前節のアルゴリズムは唯一に補うべきリレーション名や属性名を同定できるとは限らない。たとえば、図 1 に示したデータベースで、リレーション在庫の代わりにリレーション本店在庫 (商品番号, 在庫量) と支店在庫 (商品番号, 在庫量) があり、あらかじめ結合ビュー商品\_本店在庫 (商品名, 在庫量) と商品\_支店在庫 (商品名, 在庫量) が定義されているとしよう。ここで、ユーザが「商品名が Aptiva.560 の在庫量は」と聞いてきた場合 (図 13 (a) のタイプ)、FROM 句を満たすべきリレーション名 R の候補としては、結合ビュー商品\_本店在庫と商品\_支店在庫の 2 つがある。この曖昧性はユーザの発した質問文だけではどのようにそれを解析しても解消することはできない。そこで、本プロトタイプシステムではシステム主導でユーザに曖昧性解消のために合成音声で質問を発行し、それへの回答で曖昧性を消去することにした。そのアルゴリズムは前節 (1) 4 で示した T の要素数が複数の場合に当たる。前節 (1) より、現在  $S_1 = \{ \text{商品, 商品_本店在庫, 商品_支店在庫} \}$ ,  $S_2 = \{ \text{本店在庫, 支店在庫, 商品_本店在庫, 商品_支店在庫} \}$  であるので、 $S_1$  と  $S_2$  の共通集合  $T = \{ \text{商品_本店在庫, 商品_支店在庫} \}$  となる。T の要素のうち、どれが R になるのかを一意に決めるために、本来リレーション名が挿入されるべき場所にその要素を挿入した質問文を作成してユーザに問い合わせることとした。具体的には、上の例では「『商品名が Aptiva.560 の商品\_本店在庫の在庫量は』ですか」という具合である。図 13 (b) の場合もこの手法に準じている。

なお、出現語彙表を使うと前述の結合ビューを導入しなくても、より自然な日本語によるデータベースアクセスを可能とすることができるのではないかと考える向きもあろう。この考え方は理論的にはそのとおりである。しかしながら、これを実装しようとするとき、結合ビューを導入したことにより発生した曖昧性、そして本節で論じた出現語彙表の導入による曖昧性とともに、システムとユーザが音声による会話のみで解消しなければならない、それは一般には難しいのではないかと考え、そのようなアプローチはとらなかった。

```

/システム起動/
音声合成装置:「しばらくお待ちください」
/音声認識装置を起動/
音声合成装置:「質問を入力してください」
ユーザ      :「こじまの給与は」
/誤認処理のため、確認文字列を生成(入力確認モジュール)/
音声合成装置:「質問文は『こじまの所属は』ですか?
             はいかいいえで答えてください」
ユーザ      :「いいえ」
音声合成装置:「『いいえ』です。もう一度質問文を
             入力してください」
ユーザ      :「こじまの給与は」
音声合成装置:「質問文は『こじまの給与は』ですか?
             はいかいいえで答えてください」
ユーザ      :「はい」
音声合成装置:「『はい』です」
/出現語彙表を用いてのSQL文生成、Oracleでの質問処理/
音声合成装置:「結果の件数は6件です
             答えを絞りますか?」
ユーザ      :「はい」
音声合成装置:「『はい』です」
音声合成装置:「ただいまインスタンスキーの探索中です
             しばらくお待ちください」
/インスタンスキーの探索、ユーザへの問合せメッセージ生成/
音声合成装置:「社員番号は何ですか?」
ユーザ      :「わかりません」
音声合成装置:「『わかりません』です」
/次のインスタンスキーを探索/
音声合成装置:「所属は何ですか?」
ユーザ      :「営業です」
/入力確認モジュール/
音声合成装置:「あなたの入力は『所属は営業』ですか?
             はいかいいえで答えてください」
ユーザ      :「はい」
音声合成装置:「『はい』です」
             :「結果の件数は1件です
             結果、こじまさんの給与は53万円です」
             :「もう一度質問を行いますか?
             はいかいいえで答えてください」
ユーザ      :「いいえ」
音声合成装置:「『いいえ』です」
             :「これで質問を終わります」
/システム終了/

```

図14 インタクションの典型例  
Fig. 14 A typical interaction.

## 5. プロトタイプシステムの評価

### 5.1 インタクション例

以上により問題点が改良されたプロトタイプシステムを用いて、データベースアクセスを実際に行った様子を図14に示す。これは自然な日本語質問文「こじまの給与は」を入力し、出現語彙表を用いたSQL文生成を行った処理の典型例である。さらに、この質問は複数の結果を持つため、インスタンスキーを用いてのタプルの一意識別処理も行っている。

	本システム	SIMPLA[12]	IF-Kit[11]
リレーショナルDB検索機能	○	○	○
日本語文による問合せ	○	△注1	○
音声による問合せ	○	△注2	×
結果の音声出力	○	○	×
電話によるアクセス	○(可能)	×	×
マルチモーダルインタフェース	×	○	×

注1: 単語ごとに区切って入力  
注2: 単語ごとに区切って入力

図15 他システムとの比較表

Fig. 15 Comparative table with other systems.

### 5.2 他システムとの比較

以上によりプロトタイプシステムで問題となっていた同音語の問題と日本語質問文の自然性の問題が解決された。ここでは本システムを関連研究であげたシステムのうち、本システムに近いと思われる2つのシステムと比較した結果を図15に示す。評価の対象としたシステムはIF-Kit<sup>7)</sup>とSIMPLA<sup>12)</sup>である。前者は日本語文によるリレーショナルデータベースアクセスの一例として、後者は加えて音声によるアクセスが可能のため本システムと比較することが有効と考え取り上げた。電話によるアクセスが本システムでは可能とし、SIMPLAでは不可能としたのは、後者はマルチモーダルインタフェースを特徴とし、結果をグラフなどで返すことも可能であるためである。

### 5.3 音声認識装置DS200の認識率

DS200の採用しているシンタックス記述方式では認識に必要な単語をあらかじめすべて登録しておかなければならないため、大規模なデータベースを扱うと、一般的に登録単語数が増える。しかし、単語数が増えるということは認識の候補が増えるということなので、認識率は必然的に下がる。我々が開発したシステムがどれほどの実用性を有するかはDS200の音声認識能力に依存する部分もあるので、それがどの程度の認識率を有するののかの実験を行った。実験は、登録した単語が何回目の発声で正しく認識されたかを測定した。この実験に使用した語彙は、想定した応用を考慮して姓名、住所、数字の3種類とした。被験者は5人である。この結果、姓名については2回目、住所については3回目までの発声では95%の認識率を得た。数についての認識率は平均すると33%であった(1度の発声で)。しかしながら登録単語数を千個から百個に減らした場合には、平均すると80%の認識率となった。

さらに付け加えれば、DS200では登録単語の数だけ

ではなく種類によっても認識率が変化する。「この」と「おおの」など発音の似た単語や「2[に]」など語長の短い単語は判別が難しいようである。さらに、ユーザによって認識されやすい単語とそうでない単語に大きくばらつきが出た。これは音声認識装置の話者モデルに近い発音の仕方かどうかによると考えられる。したがって、ユーザの学習により話者モデルに近い発音が可能になればより高い認識率が得られる。

本システムでは DS200 が誤認した場合に対処するため、入力が正しく認識されたかどうかをユーザに確認してもらい機能を設け、最初の入力が誤認されても再度入力することを可能としている(2.2.1参照)。この機能は上記実験結果に対応するもので、ある程度の語長を持った単語群に対して十分に機能している。しかしながら、数についてはすべての数値をあらかじめ登録しておくのではなく、属性の性質に依存した限られた数値を登録することで現状は対処している。今後の音声認識技術の発達にともない、本システムの有用性がますます高まると考えられる。

## 6. 結 論

本研究ではリレーショナルデータベースへの日本語音声インタフェースの構築を考え、そのプロトタイプシステムの構築を行ってきた。市販の音声認識装置や音声合成装置を利用して構築したこのプロトタイプシステムには、同音語の音声入力による曖昧性や、より自然な日本語発話による質問という問題が従来からあったが、これらはキーやビューといったデータベースの機能の利用に加えて、新しくインスタンスキーや出現語彙表といった概念を導入することで解決できることを明らかにした。また、本研究では使用した音声認識装置の認識率を測定し、この装置をどのように使用すれば実用可能な認識率が得られるかを考察した。今後の課題としては結果出力の容量、複雑さや話者認証などがある。それらの諸問題を解決してシステムの実用性をより高めていく予定である。その結果、21世紀に向けたマルチメディア、マルチモーダル、ユービキタスなコンピューティング環境の実現に大いに貢献できると考えられる。

**謝辞** 本研究を遂行するうえでご協力や貴重なご意見を下さった様々な方々に深く感謝申し上げます。また、本研究を行うにあたり音声認識装置「DS200」とリレーショナルデータベース管理システム「Oracle7」のご提供をいただいた(株)オーヂス総研と日本オラクル(株)に心より感謝申し上げます。また本研究の一端を担ってくれた図書館情報大学増永研究室の卒業生

太田由香氏(現、日本システムウエア(株))、現在学部4年生の森田恵美子氏に深く感謝する。なお、本研究は一部平成7・8年度文部省科学研究費(課題番号07680333)および、平成9年度文部省科学研究費(課題番号08244101)の補助を得て行われたものである。

## 参 考 文 献

- 1) 荒川ゆう子, 増永良文: 音声によるデータベースアクセスの研究, 情報処理学会研究報告, 96-DBS-109 (1996).
- 2) 荒川ゆう子, 増永良文: リレーショナルデータベースへの音声アクセスインタフェースの構築実験, 第53回情報処理学会全国大会論文集(4), 3Q-6 (1996).
- 3) 藤崎哲之助, 間下浩之, 諸橋正幸, 渋谷政昭, 鷹尾洋一: データベース照会システム「ヤチマタ」と名詞句データ模型, 情報処理学会論文誌, Vol.20, No.1, pp.77-83 (1979).
- 4) 日吉茂樹, 村木一至: JAM: 日本語質問文によるデータベース検索システム, 情報処理学会データベース管理システム研究会報告, 27-4 (1981).
- 5) 中川 優, 加藤恒昭: 日本語データベース検索システムにおける意味理解方式, 情報処理学会論文誌, Vol.27, No.11, pp.1069-1076 (1986).
- 6) 牧之内顕文, 吉野利明, 泉田義男: 移行性のあるデータベース自然言語インタフェース, 情報処理学会論文誌, Vol.29, No.8, pp.749-759 (1988).
- 7) 市山俊治, 谷 幹也, 山口智治, 宮部隆夫: 日本語インタフェースキット IF-Kit, NEC 技法, Vol.47, No.8, pp.53-58 (1994).
- 8) 間瀬久雄, 木山忠博, 絹川博之: DB 検索用自然言語インタフェースにおける解釈結果確認文生成方式の開発とその評価, 情報処理学会論文誌, Vol.35, No.8, pp.1579-1590 (1994).
- 9) 笠 晃一, 小林修二, 横田将生: 日本語データベース処理システムの試作, 情報処理学会自然言語処理研究会報告, 81-6, pp.41-48 (1991).
- 10) 酒井桂一, 山田雅章, 伊藤史朗, 小森康弘, 池田裕治, 藤田 稔: 音声ガイドシステムにおける全文検索結果の利用法, 情報処理学会自然言語処理研究会報告, 95-15, pp.117-124 (1993).
- 11) 山本幹雄, 高木三功, 中川聖一: メニューによりガイドされた文節単位発話による音声対話システム, 情報処理学会論文誌, Vol.37, No.4, pp.461-470 (1996).
- 12) 有田正剛, 西村健士, 島津秀雄: マルチモーダル自然言語インタフェース SIMPLA, NEC 技法, Vol.47, No.8, pp.72-75 (1994).
- 13) (株)オーヂス総研柿本修司氏からの私信による
- 14) 荒川ゆう子, 増永良文: 音声によるデータベースアクセスの同音語問題, 電子情報通信学会第8回データ工学ワークショップ(DEWS'97)論文集, pp.227-232 (1997).

- 15) Selinger, P.G.: Access Path Selection in a Relational Database Management System, *Proc. ACM-SIGMOD 1979*, International Conference on Management of Data, pp.23-34 (1979).
- 16) Garey, M.R.: Computers and Intractability - A Guide to the Theory of NP-Completeness, W.H. Freeman and Company (1980).
- 17) 下山智明, 城谷洋司: 付録 B SUN のプロダクトと歴史, SUN システム管理, pp.797-809, アスキー出版局 (1995).

(平成 9 年 8 月 29 日受付)

(平成 10 年 2 月 2 日採録)



荒川ゆう子 (正会員)

平成 8 年図書館情報大学図書館情報学部卒業。平成 10 年図書館情報大学大学院図書館情報学研究科修士課程修了。同年富士通 (株) 入社。リレーショナルデータベースシステム

の研究・開発に従事。



増永 良文 (正会員)

昭和 45 東北大学大学院工学研究科電気および通信工学専攻博士課程修了。工学博士。同年 4 月東北大学電気通信研究所助手。昭和 58 年 4 月図書館情報大学図書館情報学部助教授。

昭和 61 年 4 月同教授, 現在に至る。その間, 昭和 50 年から 2 年間, 国際応用システム解析研究所 (IIASA, オーストリア・ウィーン) 研究員。コンピュータネットワークとデータベースの研究に従事。昭和 57 年 4 月から 1 年間, IBM サンホゼ研究所 (米国, カリフォルニア州, サンホゼ市) 客員研究員。分散型リレーショナルデータベース管理システム System R\* の研究・開発に従事。昭和 62 年度文部省派遣在外研究員としてテキサス大学オースチン校とミュンヘン工科大学に滞在。昭和 59 年から現在までマルチメディアデータベースシステム OMEGA の研究・開発に従事。著書に「リレーショナルデータベースの基礎 — データモデル編」(オーム社), 「リレーショナルデータベース入門」(サイエンス社)。監訳書に「オブジェクト指向アプローチ その全貌」(アジソン・ウエスレイ社), 「オブジェクト指向データベース入門」(共立出版)。平成 3~7 年まで情報処理学会データベースシステム研究会主査。平成 7~9 年まで情報処理学会コンピュータサイエンス領域委員長。平成 7 年より ACM SIGMOD 日本支部長。電子情報処理学会, ACM, IEEE Computer Society 各会員。