TR-IT-0296

マルチモーダル案内システムにおける対話機構およびその評価

A Dialogue Mechanism for Multimodal Guidance Systems and Its Evaluation

高橋 和子　　竹澤 寿幸
Kazuko TAKAHASHI　Toshiyuki TAKEZAWA

March, 1999

概 要

マルチモーダル案内システムにおける対話管理について述べる．インテラクティブ性の高いマルチモーダル対話システムでは一度言った語句を省略した発話が頻繁に現れる．一方，省略された語句の補完をしなければシステムはデータベース検索が行なえない．本稿では，発話履歴の新しい格納 / 検索方式を使って，省略された語句を補完する機構を提案しその形式化を行う．この方式では，現発話が示す対象を過去の対話によって言及された語句の中で，現発話に関係するすべてのものを組み込んだデータとして表し，発話ごとにそれを更新する．この方式を組み込んだマルチモーダル案内システム (MMGS) の評価実験結果についても報告し，この方式によってインテラクティブ性の高い自然な対話を行うシステムが実現できることを示す．

目次

図目次

表目次

1 はじめに

ここ数年，GUI に変わる次世代のインタフェースとして，音声，画像を使ったマルチモーダルインタフェースが注目を集めている [Tak94][KMMN94][NDI98][MT98]．マルチモーダルインタフェースを備えた対話システムは，より使いやすく親しみやすいため，カーナビを始めとする応用分野への期待も大きい．使い勝手のよいシステムを実現するためには，音声や画像の使い方，応答のタイミング等のユーザインタフェース部分はもちろん，対話システムの核部分である問題解決機構がいかに協調的な応答をするかにも大きく依存している．

これまでに実現されているマルチモーダル対話システムは，システムが質問を発しユーザがその問いに答える穴埋め方式や，道案内や操作説明のようにユーザの要求に対してシステムが長い説明をするようなシステム主導型が多かった．また，ユーザの問いに対してシステムがデータベースから答えを探し出して提供するユーザ主導型のシステムでは，ひとつひとつの質疑応答が独立しており，ユーザは質問のたびに何度も自明なことを繰り返して言う必要があった．より自然な対話を実現するためには，発話を連続性を持つものとして捉え，文脈からユーザの意図を理解したり推測したりする機構が必要である．

ATR ではマルチモーダルインタフェースを持つ観光案内システム MMGS(Multimodal/ Multimedia Guidance System) を開発した [TM98] [MT98]．このシステムは，3 次元グラフィックスで表示された地図画面上での指示や音声によって入力されたユーザの質問に対して，写真やグラフィックス上の動作および音声によってシステムが応答するユーザ主導型の質疑応答システムである (図 1)．MMGS の対話では，一度言った語句を省略した短い発話をユーザが行なう場合が多く，インタラクションが頻繁に起こるという特徴が見られる．そのため，システムがデータベースを検索するためには，省略された語句の補完や指示代名詞によって指示された対象を同定する照応の問題の解決が不可欠である．MMGS は，単純な省略・照応の解決機構を持っているが，その定義はかなり *ad hoc* なものになっていて拡張性に乏しい．本稿では，MMGS の問題解決部における省略・照応の解決機構に焦点をあて，よりインタラクティブな対話を実現する仕組みを提案する．

MMGS の一つの場面である関西学研都市光台地区の図を見ながらの対話を考えてみよう．

U1: これはなんですか [画面上を○で囲む].
S1: 「けいはんなプラザ」です.
U2: 喫茶店はありますか.
S2: 「ライン」があります [写真を見せる].
U3: ATR にはありますか.
S3: カフェテリアがあります [写真を見せる].

[ ] 内は画面上の操作を示す．U はユーザの発話，S はシステムの発話をそれぞれ表す．

ユーザの発話で U2 では「けいはんなプラザに」が，U3 では「喫茶店は」がそれぞれ省略されている．システムがデータベースを検索するためには，これらの省略を補わねばならない．一方，もし U3 で

U3': 広いですか.

という質問がきたならば，省略されているのは「ラインは」である．「喫茶店」も「ライン」も名詞であり，ともに喫茶店という範疇でとらえられるため，文法や意味的な制約からはどちらが適切か判断できない．

一般に，省略・照応が何をさすかを判定するためには，過去の発話履歴を列としてとらえ，それらを木構造やスタックとして記憶し，それを参照して決定するという方法がとられている．そのため発話の継続性が高いと，履歴の探索に時間を要するという問題点がある．

本稿ではこれを解決するために，履歴を格納 / 検索する新しい方法を提案する．すなわち，これまでの対話によって言及された語句の中で，現発話に関係するすべてのものを **Q 話題 / A 話題**という一つのデータの中に組み込み，発話ごとにそれを更新していくという方法である．このデータはこれまでの話題の遷移を反映したものになっているので，履歴の探索時に木やスタックを辿る必要がなく，省略されたものや指示されるものが速く発見できる．

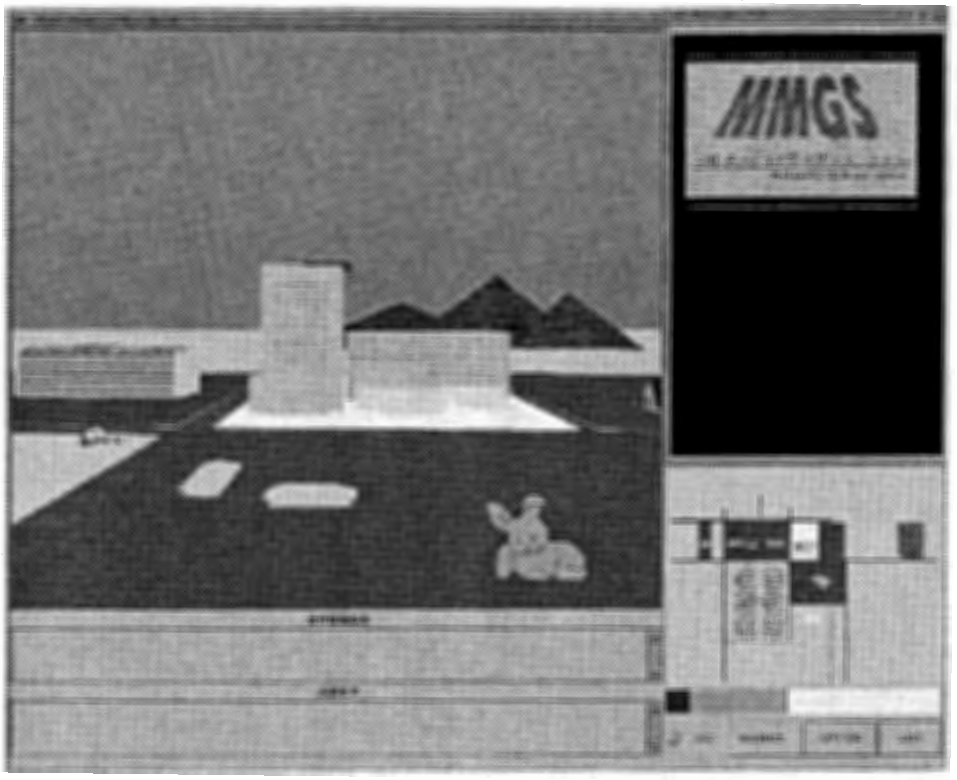

図 1: MMGS の表示例

本稿は以下のように構成される．まず，2章で MMGS の概要を説明し，3章で Q 話題を使った省略補完について述べる．4章で他の手法との比較について論じ，5章で評価実験結果について述べる．最後に6章で結論を述べる．

2　システム概要

2.1　システム概要

図 2に MMGS のシステム構成を示す．ユーザからの入力は音声および画面上の指示による入力を認識する認識部を通った後，意味解析部によって意味解析の行われた意味表現が生成される．問題解決部ではこの意味表現と履歴データに基づいてデータベースの検索を行ない，その結果から画面の更新情報や応答文を生成し，さらに履歴データを更新する．そして，それを音声合成して出力したり 3 次元グラフィックスや写真によって表示する．画像出力では SGI のシーン表示機能を用いて 3 次元図形を表示する．また，画面を回転させて，話題となっているオブジェクトが最もよく見える位置まで視点を移動させる．応答時には，応答文を合成音声で出力させるとともに，対象オブジェクトを点滅させる．遠方にあって画面上表示されていないオブジェクトに対する質問も可能であり，回答では矢印表示などを利用した出力を行なう．

データベースは階層型オブジェクト構造を持っており，建物，概念などはオブジェクトとして定義される．各オブジェクトは属性として意味情報と空間・形状情報を合わせて持つ．意味情報には観光スポットの場合は見学可能な時間帯や料金など，集合ビルの場合は中に含まれる施設などが含まれる．空間・形状情報は地図上の位置，色，形状，材質などである．

また，応答が返るまでの待ち時間を楽しませるためにキャラクタアニメーションを画面上に出す (図 1)．このキャラクタは，以下の 4 フェーズから成り，各フェーズは数個のアニメーションパターンから成る．

1. 初期状態 (寝ている)
2. 音声認識時 (目をあけ耳をピクピク動かす)

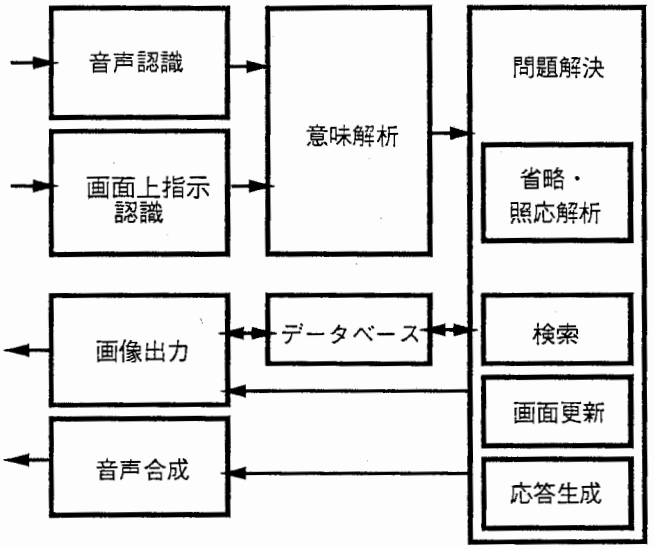

図 2: MMGS のシステム構成

3. 応答出力時 (立ち上がって画像を案内する)

4. 応答出力終了 (あくびをして寝る)

2.2 意味表現

音声認識部では，文脈自由文法および前終端記号の bigram を使った言語モデルを使っている．文脈自由文法で書かれた各文法規則には素性構造に基づく意味解釈規則が付加されている．音声認識部ではこの文法を用いて HMM-LR 手法による認識を行ない，統語解析木構造を出力する．意味解析部では，この木構造から対応する意味解釈規則を取出し，対応する素性構造間で単一化処理を行う．その結果，一発話全体に対する意味表現を決定して出力する． MMGS の特徴の一つはこのように音声認識部と言語解析部が文法規則として同じものを使用していることであり，これによって効率向上をはかっている．

問題解決部が意味解析部から受け取る意味表現の定義を以下に示す．

```
sentence::= cls | cls conj sentence
cls      ::= (class,action,object)
object   ::= (label,ftr_1,...,ftr_n)
class ::= yn-q | which | request | ...
action   ::= be | exist | change | ...
label ::= transfer | restaurant | ...
ftr      ::= (fn,fv)
fn       ::= goal | loct | spec | ...
fv       ::= ATOM | ftr
```

発話文が単文の場合，その意味表現は $(class, action, object)$ の 3 項組になる．複文や重文の場合はこの構造が接続詞 $(conj)$ で連結されたものになる． $class$ は質問の種類を， $action$ は主動作の種類をそれぞれ表す． $object$ はその文で言及されている対象物を表し，名前 $(label)$ および素性 (ftr) の n 個の $(n = 0$ を含む) 有限集合であり，その構造は $class, action, label$ によって決まる．各素性は素性名 (fn) と素性値 (fv) から成る．

素性名は $object$ を表すのに必要な性質の種類を表す．たとえば，移動 $(transfer)$ という $label$ を持つ $object$ に対しては起点 $(sour)$ と終点 $(goal)$ と手段 $(mann)$ という素性名を持つ構造がつくられる． $spec$ は修飾句を表す素性名である．素性値は対応する素性名がその発話でとる値であり， $ATOM$ は具体値がはいることを表す．

以下に例を示す.

例

「近鉄奈良駅から奈良県新公会堂まで歩いていけますか」

```
( yn-q,
  exist,
  ( transfer,
      (sour, kintetsu-nara-station)
      (goal, nara-public-hall)
      (mann, walking) ))
```

この例では, yn-q が *class*, exist が *action*, (transfer ...) が *object* にそれぞれ相当する. 主語や目的語が省略された発話に対しては, 素性名ができても素性値が決まらない (すなわち変数のままである) 構造ができる. 以後, 値の決まっていない部分は '?' で始まる文字列で表す.

3 インタラクション機構

MMGS は基本的に質疑応答システムであり, 対話はユーザの質問とシステムの回答という隣接ペアの列と考えられる[1]. ユーザの質問に対して **Q 話題**を, システムの回答に対して **A 話題**をそれぞれ対応づける.

3.1 Q 話題の生成

簡単のため, 各発話は単文 1 つで話者交代が生じるものと仮定する.

音声認識が正しく行われるとすると, $class, action$ が省略されることはないので, 意味表現として $(class, action, object)$ という構造が得られる. ただし, *object* 自身やその中身は部分的に省略されている可能性がある. *object* の内容を Obj とし, 直前の隣接ペアに対応する A 話題を $Ap = (label, (fn_1, fv_1), \ldots, (fn_k, fv_k))$ とする. Ap を使って Obj の省略補完を行い, 現在のユーザの発話に対する Q 話題 $Qinfo$ を生成する方法を以下に述べる.

[Q 話題の生成方法]

初期状態で特定の地図が表示されていない時は, $Qinfo$ の初期値は ϕ(情報なし) である.

Obj が変数を含まなければ, $Qinfo = Obj$ とする.

Obj が変数を含む時は以下の処理を行なう.

[1] *label* が変数でない場合

この時, Obj の構造は決定しているが, その中で素性値が変数であるような素性が存在する. 簡単のため, そのような素性がただ一つ存在すると仮定し, それを $(fn, ?fv)$ とおく.

(1) Ap が素性 (fn, fv) を含めば, Obj 中の $?fv$ に fv を代入したものを $Qinfo$ とする.

(2) 含まなければ, データベースから *label* をもつオブジェクトを取り出し, 属性 fn に対応する属性値を fv' とする. $fv' = Ap$ ならば, Obj 中の $?fv$ に Ap を代入したものを $Qinfo$ とする.

(3) Ap が (fn', fv') (ただし $fn \neq fn'$) を含めば, Obj 中の $?fv$ に fv' を代入したものを $Qinfo$ とする.

(4) いずれでもない場合は, ユーザに聞き返す.

[1] 回答を特定できない場合は, 不明箇所を明らかにするための質問をシステムが行なう可能性がある. この場合はユーザの最初の質問からそれに対するシステムの回答までを一つのユニットと考える. また, 回答は発話だけでなく, 対象物の画面上の点滅を伴う場合もあれば, 画面切り替えになる場合もある.

[2] $label$ が変数の場合

(1) Obj の素性がまったく判明していない場合

(a) Ap から具体例 $instance$ にポインタがはられている場合は(ポインタについては次節で説明する), $Qinfo = instance$ とする.

(b) それ以外なら, $Qinfo = Ap$ とする.

(2) Obj の素性が少なくとも一つは判明している場合

(a) 素性名, 素性値ともに判明している場合

簡単のため, そのような素性がただ一つ存在すると仮定し, それを (fn, fv) とおく.

(i) $fn = fn_i$ である i $(i=1,\ldots,k)$ が存在する時

Ap 中の fv_i を fv で置き換えたものを $Qinfo$ とする.

(ii) $fn = fn_i$ である i $(i=1,\ldots,k)$ が存在しない時

データベースに $(label, (fn, fv), (fn_1, fv_1), \ldots, (fn_k, fv_k))$ というオブジェクトがあれば, それを $Qinfo$ とする. なければ, データベースの中で属性 fn をもち, その値が fv_i または Ap であるようなオブジェクトをすべての $i(i = 1,\ldots,k)$ について検索し, 処理 $(*)$ を行なう.

(b) 素性名のみ判明していて素性値が変数の場合

簡単のため, そのような素性がただ一つ存在すると仮定し, それを $(fn, ?fv)$ とおく.

(i) $fn = fn_i$ である i $(i=1,\ldots,k)$ が存在する時

データベースの中で属性 fn をもち, 値が fv_i であるようなオブジェクトを検索し, 処理 $(*)$ を行なう.

(ii) $fn = fn_i$ である i $(i=1,\ldots,k)$ が存在しない時

データベースの中で属性 fn をもち, その値が fv_i または Ap であるようなオブジェクトをすべての $i(i = 1,\ldots,k)$ について検索し, 処理 $(*)$ を行なう.

処理 $(*)$

対応するオブジェクトが唯一見つかれば, その名前を lbl とする. 複数見つかれば, それらの名前を列挙してユーザにどれをさすのか聞き返しその答を lbl とおく. その結果 $Qinfo = (lbl, (fn, fv_i))$ とする.

以上では省略について述べたが, 照応に関しても同様に扱える.

このようにして生成された $Qinfo$ は, これまでの対話によって言及された対象の中で, 現発話に関係するものを必要十分に含む. $Qinfo$ の生成の際, 唯一履歴として参照にするのが Ap だが, この参照では語句の長さや構造にかかわらず一度のアクセスによって補完すべき語句を見つけることができる. また, これまでの発話履歴を列として持たず, すべてを組み込んだ最新情報という形で一つ持っているだけなので, 探索も速い.

3.2 A 話題の生成

ユーザの質問に対応した Q 話題 $(Qinfo)$ が生成されると, システムはこれをもとにデータベースを検索する. その結果, Q 話題に追加, 訂正などの修正を加えた A 話題 $(Ainfo)$ を生成し, これを適当な形に加工してユーザへの回答とする.

ユーザの発話を 6 種類に分け, それぞれの場合について $Qinfo$ から $Ainfo$ を生成する方法を以下に述べる.

[A 話題の生成方法]

(1) wh-question によるオブジェクトの属性値の質問

例「(奈良駅から新公会堂までバスで) どのくらいかかりますか」

システムは $Qinfo$ に対応するオブジェクトをデータベースから探す．存在する場合は，その属性値が返されるので，$Qinfo$ に対応する素性を追加したものを $Ainfo$ とする．存在しない場合は，$Ainfo = Qinfo$ とする．

(2) yn-question によるオブジェクトの存在の質問

例「(奈良公園の中に) レストランはありますか」

存在する場合は，システムは $Qinfo$ に対する具体例を返すので，$Qinfo$ からその具体例へのポインタをはったものを $Ainfo$ とする．存在しない場合は，$Ainfo = Qinfo$ とする．

(3) yn-question によるオブジェクトやその属性の同定

例「これは ATR ですか」

答が yes なら，$Ainfo = Qinfo$. no の場合は，$Qinfo$ の中で訂正部分に対応する素性値を変更し，それを $Ainfo$ とする．

(4) 操作要求

例「奈良周辺の地図に切り替えて下さい」

$Ainfo = Qinfo$.

(5) システムの同定質問に対する回答

例 (「見学時間を教えて下さい」「大仏のですか興福寺のですか」に続く発話として)「大仏のです」

照応・省略の解決過程の一部としてとらえることができる．

(6) 繰り返しの要求，礼

例「どうもありがとう」

$Qinfo, Ainfo$ には影響をおよぼさない．

3.3 音声認識誤りへの応用

音声認識がうまく行われなかった場合でも意味表現として，$class, action$ さえ認識できていれば，3.1節，3.2節で述べた方法によって，不明な部分の補完が可能である．

更に，MMGS では $class$ または $action$ が決定できない時は，聞き取れた断片だけを取り上げてユーザに問い合わせるという手法をとっている．たとえば，画面上の指示入力によって特定の建物が指示されたことがわかれば，「それは奈良県新公会堂ですが，どんな用件でしょうか」などと聞き返す．

3.4 動作例

3.1節で示した Q 話題の生成方法のいくつかの場合について応用例を示す．

現在のユーザの発話に対する意味表現を Rep，Q 話題を $Qinfo$，直前の隣接ペアに対応する A 話題を Ap で表す．

[1] の例

「時間はどのくらいかかりますか」

```
Rep =
  (how-long,
    need,
      (time,
        (spec, ?SPEC))
```

直前の発話で「奈良から京都まで電車で行く」という言及がなされてものとする.

```
Ap =
  (transfer,
    (sour, nara),
    (goal, kyoto),
    (mann, train))
```

Ap に修飾語を表す *spec* という素性名は存在しないので, *Ap* を候補として *?SPEC* に代入する. データベースを検索すると, 対応するオブジェクトが存在するので, これを *Qinfo* として確定する. その結果, この質問は「奈良から京都まで電車で行く時間はどのくらいかかりますか」と解釈される.

```
Qinfo =
  (time,
    (spec,
      (transfer,
        (sour, nara),
        (goal, kyoto),
        (mann, train))))
```

[2](1)(a) の例

「広いですか」

```
Rep =
  (yn-q,
     wide,
       ?OBJECT)
```

直前の発話で「けいはんなプラザの喫茶店」が言及されているものとする.

```
Ap =
  (coffee-shop,
    (loct, keihanna-plaza))
```

また, ここから rhein へポインタがはられているものとする.

?OBJECT の構造には制約がないので, *Ap* そのものを代入してよいが, *Ap* からポインタがはられているので, その先の *rhein* を代入し,

```
Qinfo = rhein
```

を得る.

[2](2)(a)(i) の例

「ATR にはありますか」

```
Rep=
  (yn-q,
    exist,
      (?LABEL, (loct, ATR), ?FEATURE))
```

直前の発話で「けいはんなプラザの喫茶店」が言及されているものとする.

```
Ap =
  (coffee-shop,
    (loct, keihanna-plaza))
```

また，ここからrheinへポインタがはられているものとする．

Ap には *loct* という素性名があるので，この素性値を置き換えて，

```
Qinfo = (coffee-shop, (loct, ATR))
```

を得る．

[2](2)(a)(ii) および [1](3) の例

「ここから京都までどのくらいかかりますか」

```
Rep=
  (how-long,
    need,
      (?LABEL,
        (sour, HERE),
        (goal, kyoto),
         ?FEATURE))
```

ただし，$?FEATURE$ は素性の有限列である．

直前の発話で「奈良公園の土産物屋」が言及されているものとする．

```
Ap =
  (souvenior-shop,
    (loct, nara-park))
```

Ap には *sour*, *goal* という素性名がないので，データベースで *sour*, *goal* という属性を持つオブジェクトをさがす．すると，唯一交通手段 (*transfer*) というオブジェクトが発見される．$?LABEL$ に *transfer* が代入されると，以下の構造が決まり，

```
(transfer,
  (sour, HERE),
  (goal, kyoto),
  (mann, ?MANN))
```

となる．次に，指示代名詞 $HERE$ の指示対象を決定する．*Ap* には *sour* 素性を持つものはないので *Ap* 自体を代入する．すると，

```
(souvenior-shop,
  (loct, nara-park))
```

が得られるが，これに対応するオブジェクトはデータベースにないので，次候補として

```
nara-park
```

をとってデータベースを検索する．すると，「奈良 (公園) から京都までの電車による移動」というオブジェクトが得られる．この場合は交通手段も同時に決定される．

```
Qinfo=
  (transfer,
    (sour, nara-park)
    (goal, kyoto)
    (mann, train))
```

この例では，前半部の構造を決める部分が 3.1節の [2](2)(a)(ii) に，後半部の指示代名詞の指示対象の決定部分が [1](3) の場合にそれぞれ対応する．

4 他の手法との比較

一般に省略・照応の解決は非常に難しい問題であり，これを完全に実現したシステムは今のところない．

照応解決の代表的な理論ベースとしては，centering 理論に基づくもの [WIC94][GJW95] や談話表示理論 [Kam81][Hei82] などがある．

centering 理論は，ランク付けされた談話内容 (discource entity) の集合を発話ごとに格納し，各集合でもっとも重要なものが指示代名詞によって指示されるとする考え方である．この手法ではランク付の妥当性が問題となる．

中心となる話題を木構造として格納し，それを参照することによって省略や照応の解決をはかる方法も提案されている．しかし，この方法では表層表現から話題を決定する方法の有効性が問題になる他，どこまで話題を遡るかを考えなければならないという問題が生じる．

本稿で提案した手法では，直前の隣接ペアの A 話題にこれまでの履歴も含めた情報が格納されているため，直前の A 話題のみを参照すればよく，話題の切り替え等を気にしなくてよい．もし，ひとつの話題に対する発話列が長く続けば，Q 話題 /A 話題の素性が増加していくが，一般にマルチモーダル対話システムを使った対話では，ある程度話題が継続すれば他の話題に移るものが多く，Q 話題や A 話題は一定の大きさを越えることはほとんどないと思われる．

談話表示理論は発話ごとに指示されるものを変数で表し，発話間の変数の束縛という形で照応解決しようというものである．この理論に基づく形式的体系は対話の解析には有効だが，実際のシステムに組み込んでそれを実時間応答をさせるにはかなり無理がある．Q 話題 /A 話題はこれまでの話題の遷移まで反映したものになっているので，履歴の探索時に木やスタックを辿る必要がなく，省略されたものや指示されるものが速く発見できる．

省略に関してはこれまでも自然言語インタフェースを持つ多くの対話システムが取り込んでいる [GAMP87][DSP91]．これらの多くは文法的制約と意味的制約に基づいて省略された語句を決定し，それでも候補が絞れない場合には語句ごとに意味の近さを定義して文中の他の語との共起関係から決定するという方法をとっている．しかし，本稿の冒頭の例のように，省略されているものが「喫茶店」と「けいはんなプラザの中の喫茶店 (ライン)」かを判定しようとすると，語句の意味の近さをかなり細かく定義せざるをえない．その結果，データベースの語彙を増やす際に非常に煩雑な改変作業が必要になる．本論文で提案した手法では，語句の近さは定義する必要がないので，データベースの語彙を増やすのはたやすい．

5 評価実験

5.1 実験内容

5.1.1 目的

第 3章で述べたインタラクション機構を MMGS の問題解決部に組み込み，被験者を使った評価実験を行なった．実験では以下の項目について評価することを目的とした．

- 音声入出力について
  認識の精度，速度 (すべての処理)，操作性
- 画像入出力について
  動きの妥当性，わかりやすさ，操作性
- インタラクション機構について
  対話の流れ，応答文の妥当性，照応・省略解決機構の妥当性
- 全体
  応答時間，待ち時間の印象，システム全体

(秒)

ステップ	全体 (275 発話)	認識成功時 (238 発話)	失敗時 (37 発話)
1. 発話終了から認識結果出力まで	3.1	3.0	3.6
2. 認識結果出力から問題解決部入力まで	3.4	-	-
3. 問題解決部入力から応答実行開始まで	0.4	-	-
応答時間 (合計)	6.9	6.8	7.5

表 1: 応答時間 (平均値)

5.1.2 方法

システム概要と使用方法を説明した後，被験者に例文による対話および課題付自由対話を行なってもらい，ログファイルをとった．話者適応やシステムの使い方の練習，音声入力の練習はいずれも行なっていない．また，被験者には実験終了後アンケートを記入してもらった．

例文による対話では，2 場面 15 文の発話を行なってもらった．自由対話では，課題を一つ指定し，10 分程度の時間内で課題に即した対話をしてもらった．原則として言い回しの制限はしないが，あまりに認識が悪い場合は適宜誘導することにした．

5.2 結果および考察

5.2.1 動作環境

MMGS は SGI Octane および Linux/PC Endeaver Pro300 上に実装されている．前者には音声画像処理部分，後者にはその他の処理部分が実装され，両者がネットワークを介して通信している．認識 / 解析辞書サイズは約 300 語，オブジェクトデータ数は約 50 個である．

5.2.2 被験者内訳

被験者は 19 人 (男 11 人，女 8 人) である．年代は 10 代～ 40 代，職業は学生，大学教官，研究開発従事者であり，全員コンピュータの経験を持っていた．また，専門家を含め対話システムに何かしらの興味を持つ者が半数程度いた．さらに，例文対話実験のみに参加した人が一人いたため[2]，以下の集計では例文対話のログデータは 20 名分になっている．

5.2.3 システムログ結果

例文対話

例文対話については応答時間，音声認識率，繰り返し質問の回数を測定した．評価実験の所用時間は 1 名あたり約 10 分であった．

ユーザには 1 発話ごとにマイクを切断するように指示し，そのタイミングを発話終了時刻とした．切り忘れやあまりにタイミングの遅いものは集計から除外している[3]．発話終了時刻からシステムの応答開始時刻までを応答時間とする．

応答時間および各処理過程の実行時間の平均値を表 1に示す．

平均応答時間は 7 秒弱であり，ユーザには少々遅いという印象を与えている．最大の原因は認識結果をテキストデータとして SGI マシンからネットワークを介して PC に送っていることである．これは，意味解析部で使用したプログラムがもともと PC 上に LISP で開発された汎用意味解析モジュールだったためであり，そのためステップ 2 には意味解析結果を MMGS のデータベースへの問い合わせ用に書き換える

[2]ちなみに，この人は外国人女性で，ある程度日本語が話せるぐらいのレベルであったが，認識率に関しては他の被験者とほとんど変わらない結果が出た．

[3]表 1の発話数と表 2の発話数が異なるのはこのためである．

	被験者数	発話数	正解認識数	認識率 (%)
男性	11	185	157	84.9
女性	9	145	132	91.0
合計	20	330	289	87.6

表 2: 音声認識率

場面 1 [光台地区の地図]		繰返し質問回数
1	ATR はどれですか	2
2	レストランはありますか	0
3	高の原駅まで歩いていけますか	3
4	時間はどのくらいかかりますか	2
5	奈良の地図に切り替えて下さい	1
場面 2 [奈良の地図]		
6	奈良公園の観光コースを教えてください	3
7	時間はどのくらいかかりますか	2
8	大仏は何時から見られますか	5
9	いくらかかりますか	1
10	近くで昼食は食べられますか	1
11	奈良公園の中にはないんですか	2
12	大仏殿から歩いて行けますか	7
13	これは何ですか	2
14	奈良のみやげものを教えてください	1
15	どうもありがとう	0
合 計		32

表 3: 繰り返し質問回数

という過程も含まれており，これらが実行時のオーバーヘッドになった．ステップ 2 の処理自体は 0.1 秒もかからないと予測され，意味解析部を MMGS 専用のものに修正し同一マシン上で実装することによって 1 桁以上の高速化が期待される．また，第 1 ステップは，現在 150 万幅で行なっている認識のビーム幅を減少すれば処理速度は半減可能なことがわかっている．第 3 ステップでは，かなりの数に場合分けして応答生成を行なうため，オーダをさげるほどの大きな効率アップはのぞめないが，場合分けの順序を工夫したり応答パターン数を増やすことで，ある程度の高速化は可能である．以上によって，全体の応答時間は少なくとも 3 分の 1 程度に削減することが期待できる．

次に，表 2に音声認識結果を，表 3に繰り返し質問回数を示す．認識率は 90% 弱であり，前回の予備実験で 70% 程度だった [TM98] ことと比較すると，かなり認識率はあがったといえる．繰り返し質問数は例題対話については非常に少なく，大体問題なく認識されたといえる．強いてあげれば「大仏」が認識されにくいようであった．繰り返しの原因はすべて音声認識の失敗である．また，音声認識が失敗しているにもかかわらず，以後の処理の結果が同一になったことで正しく応答が返り，繰り返し発話が生じなかった場合がある[4]．

自由対話

評価実験の所用時間は 1 名あたり約 10 分であった．

表 4に，各被験者の 1 セッションにかかった時間，発話数などの測定結果を示す[5]．各項目は以下の通りである．

[4] そのため，繰り返し質問回数の合計値が表 2の音声認識失敗数と異なっている．

[5] 被験者 A はログ不良のため無視した．

項目 被験者	1	2	2a	2b	3	3a	4	5 sec	6 %	7 %
B	15	8	4	5	6	6	0	373	53	63
C	17	8	2,3	2	5	4	0	304	47	25
D	19	6	2	1	1	1	0	412	32	17
E	9	6	3	1	4	4	0	218	67	17
F	19	7	5	4	6	5	0	525	37	57
G	37	18	6,3,2,2,2	5	15	15	3	729	49	28
H	28	23	2,2,4,14	12	20	12	0	737	82	52
I	35	11	2,3,2	2	10	8	1	686	31	18
J	31	11	4	5	10	8	0	744	35	45
K	38	14	0	7	7	7	0	917	37	50
L	23	14	2,3	3	5	4	0	578	61	21
M	14	7	3	0	6	6	0	324	50	0
N	32	14	2,2,2	3	7	5	1	825	44	21
O	26	4	0	0	3	2	0	594	15	0
P	47	19	2,2,3	7	8	6	0	832	40	37
Q	32	8	3,2	3	6	3	0	708	25	38
R	26	11	2,3,3	5	8	6	0	565	42	45
S	35	11	2	4	7	5	0	848	31	36

表 4: 自由対話のシステムログ

1. 1セッションの発話数
2. 正しく認識された発話数
2a. このうち連続出現するものの連続数
2b. このうち省略・照応のある発話数
3. 成功した質疑応答組数
3a. このうち異なる種類数
4. 画面上指示の回数
5. セッションの時間 (システムダウンしている時間を削除)
6. 文認識率 (= 項目 2 / 項目 1)
7. 省略・照応の出現頻度 (= 項目 2b / 項目 2)

A を除く被験者の平均セッション時間は 10 分 6 秒，平均発話数は 26.8 であった．また，有効発話数 389 の平均応答時間は 7.6 秒，平均認識時間は 3.8 秒，平均文認識率は 41.3% であった．

項目 2a は正しく認識された発話，つまり質疑応答が成功した隣接ペアがどれだけ連続して見られたかを示したものである．たとえば，被験者 C では，2 個連続してうまく応答された発話列と 3 個連続してうまく応答された発話列があった．この項目の結果が示すように，ほとんどの被験者が (おそらく無意識のうちに) 省略・照応を使った発話をしておりこれらの解決を当然あるべき機能と考えている．従って，このようなマルチモーダル対話システムで自然な対話の流れをつくるためには，省略・照応の解決が不可欠であることが例証された．

次に，応答処理失敗の原因を表 5に示す．

この表で，認識誤り・文法未対応は認識側の失敗によるものであり，認識できない文型，認識誤り，ノイズ，ポーズのために複数の文と認識されたことなどが含まれる．

文型未対応・データ不足には，データベースにデータが無い，問題解決部で対応できない文型が入力されたことなどが含まれる．

履歴管理ロジック・アルゴリズムの不備では，認識の失敗によって履歴が変わり，正しい情報を与える

失敗原因	件数	割合 (%)
認識誤り・文法未対応	129	62.0
文型未対応・データ不足	58	27.9
履歴管理ロジック・アルゴリズムの不備	16	7.7
解析結果書き換え	5	2.4
合 計	208	100.0

表 5: 応答処理失敗の原因

ことができなかったことによるものが多かった．音声対話では認識失敗は当然起こるものであり，今後は認識失敗に対しても対応できるような履歴管理が行なえるようにアルゴリズムを改良していく必要がある．また，本稿で提案した履歴管理の手法では直前の Q 話題 /A 話題のみを見ればよいことになっているが，ユーザの発話によっては，それでは不十分なものもあった．たとえば，以下のような場合である．

U1:「近鉄奈良駅はどこですか」
S1:「近鉄奈良駅はこちらです」[画面上表示]
U2:「奈良県新公会堂はどこですか」
S2:「奈良県新公会堂はこちらです」[画面上表示]
U3:「行き方を教えて下さい」

U はユーザの発話，S はシステムの発話をそれぞれ表す．この場合，U3 は「近鉄奈良駅から新公会堂までの行き方」をさしており，「近鉄奈良駅」と「奈良県新公会堂」の両方を履歴として覚えておく必要がある．しかし，本手法では U2 で既に話題は「奈良県新公会堂」に移っていると判断し，「近鉄奈良駅」の参照は不可能である．これは本手法の限界ともいえるが，今回の実験でこのような対話が行なわれた原因として被験者がなんとしても情報獲得に成功したいという意図から認識されそうな文を選んで発話したと予測される．認識率が向上し，データベースが十分な量になった場合に果たしてこのような発話をユーザが行なうかについては疑問が残る．

解析結果の書き換えには，ある種の質問に対して書き換え部の処理が未対応であったことなどが含まれる．前節でも述べたように意味解析部を MMGS 専用のものにカスタマイズすれば最終的にこの書き換え処理は不要になる．

また表示に関しては，話題となっている対象物が見えやすい位置に画面が移動する機能があり，マウス操作によってユーザ自ら視点を動かすこともできるのだが，実際に画像を動かした者が 3 人，ポインティングをした者が 3 人のみだった．これは，どの被験者も音声入力に手一杯で，画像入力まではほとんどできなかったためと考えられる．

課題に画像入力を特に必要とするものがなかったのも原因の一つだが，このようなシステムの「マルチモーダル入力」を使いこなすためには，音声認識率の向上および問題解決部の補強が必須である．

5.2.4 アンケート集計結果および考察

アンケートの集計結果を以下に示す．

- 音声入出力について
  例文対話に関してはおおむね好印象であったが，課題付き自由対話についてはほとんどの人が表現したいことが伝えられずストレスを感じたようである．

- 画像入出力について
  動きの速さ，大きさなどは特にわかりやすい，わかりにくいという意見はなかった．この設問は，比較対象がないため答えにくかったようである．

- インタラクション機構について
  課題付き自由対話では全員がタスク達成できず，認識で失敗していたのがほとんどであった．そのため話しにくいという印象が濃い．応答文の表現についてはそこそこの評価を得た．

- 全体
  応答時間は若干長めという意見が多かった．これについては5.2.3節で示したように高速化可能であり，今後の課題である．また，うさぎのアニメーションキャラクタは好評でありバリエーションがほしいという意見も多かった．

  さらに，マルチモーダルインタフェースを持つシステムの将来性についてだが，19名中15名が案内システムとして適当な形態だと回答している(残りの4名はどちらともいえないと回答)．また，19名中17名が使ってみたいと回答し，

  そのための条件として音声認識率の向上を条件にあげた者が最も多かった(14名)．使いたくないという意見の中には，セキュリティを心配する意見と，機械に向かって話しかけることへの抵抗があった．後者については，機械に向かって話しかけるイメージをもっと前向きなものにする必要がある．しかし，そのためにはできるだけ人間的なシステムをつくるのが最適なのか，それともももっと別の形で実現するのか，これは今後のユーザインタフェースの研究の課題である．

5.3 インタラクティブ性の評価と実験方法

対話システムの評価，特にインタラクティブ性の評価に関しては，基準となる指標が今のところ存在しない．有目的対話では，タスク達成率がひとつの指標となるが，対話システムが(分野は限られても)十分実用に耐えるほど鍛えられていない場合は，初心者のタスク達成率は極端に低くなってしまう．

そのため，[NDI98]では，話者適応とともに練習セッションをもうけて被験者がある程度システムを使いこなせるようになってから実験を開始している．このような方法だと被験者のストレスも少ないし良いデータが得られるが，一方，使い方を限定してしまい，型にはまった言い回しが多くなって自然な対話の流れをつくるという目的からはずれてしまうという欠点もある．

また，マルチモーダル対話システムから対話管理部分のみを切り出すのは難しいことから，音声認識を切り離しマウスによるメニュー選択やキーボードからのテキスト入力によって対話を行ない，問題解決部分のみの評価データを測定することも考えられる．しかし，この場合入力が音声・画像インタフェースを通した時のものと同じものになるかという点では大きな疑問がある．

マルチモーダル対話システムの評価実験方法については我々が今回とった方法が最良とはいえず，今後も検討が必要であろう．

我々はインタラクティブ性の指標のひとつとして，以下のものを考えた．

- 正しく認識された発話の中で連続出現する質疑応答ペアの数
  多いほどインタラクティブである

- 省略・照応の数
  多いほどインタラクティブである

実験結果では，双方ともかなりの数が見られ，これらを一つの指標としてよいと思われる．

Litmanらは，対話システム全体の性能評価として，話者交代の回数，質問の種類，アンケートによるユーザの満足度などをパラメータとして計算する方法を使い，[LPW98]で応答戦略の異なる二つのシステムについて評価実験を行なった．しかし，彼らが実験に用いたのはいずれもシステム主導型の対話システムであり，MMGSのようにユーザが主体的に発話を行なうような場合にはそのままでは当てはまらない．

6 おわりに

マルチモーダル対話システムにおける対話ではインタラクションが頻繁に生じ，省略表現が多く見られる．本稿では，過去の対話によって言及された語句の中で，現発話に関係するすべてのものを組み込んだデータ**Q話題/A話題**として現発話の示す対象を表し，これを使って省略されたものを補完する機構を提案しその形式化を行った．この手法によって，インテラクティブ性の高い自然な対話を行なうシステムが実現できる．さらに，この手法を対話システムMMGSに組み込んで評価実験を行なった．

この手法の特徴として以下のものがあげられる．

- ドメインに依存せず，一般のマルチモーダル対話システムに適応可能である
- 実時間応答が可能である
- データベースの語彙を増加させるのが比較的簡単である

なお，本システムは光台地区，奈良地区の場面に対応しているが，両者の語彙数や画像データ量には大きな差はない．そのため，問題解決部にかぎっていえば両者の処理速度の差はほとんどない．異なるドメインに適用した場合，データにそれなりの属性を用意する必要があり，データベース探索に時間がかかる可能性がある．しかし，「インタラクティブ性」という観点からはデータ量よりもむしろ発話の継続性と処理速度の関係を考慮すべきである．今後は実験結果をシステムにフィードバックするとともに，評価実験方法特にインタラクティブ性の評価方法について検討していく予定である．

謝辞

MMGS の実装に関しては田中吾郎氏，塩崎純氏，馬場武雄氏に御協力いただきました．ここに記して感謝します．

参考文献

[DSP91] M. Dalrymple, S. Shieber and F. Pereira. Ellipsis and Higher-Order Unification, Linguistics and Philosophy Vol.14, No.4, 1991.

[GAMP87] B. J. Grosz, D. Appelt, P. Martin and F. Pereira. TEAM: An Experiment in the Design of Transportable Natural Language Interfacses, Artificial Intelligence, vol.32, no.2, pp.173–244, 1987.

[GJW95] B. J. Grosz and A. K. Joshi and S. Weinstein. Centering: A Framework for Modeling the Local Coherence of Discourse, Computational Linguistics, vol.21, no.2, pp.203–225, 1995.

[Hei82] I. Heim. The Semantics of Definite and Indefinite Noun Phrases , *U.Massachusetts*, 1982.

[Kam81] H. Kamp. A Theory of Truth and Semantic Representation, In J. A. G.Groenendijk, T. M. V. Janssen and M. B. J. Stokhof (eds.) *Formal Methods in the Study of Language* pp.277–322, 1981, Mathematical Centre Tract 135, Amsterdam.

[KMMN94] 神尾広幸, 松浦博, 正井康之, 新田恒雄. マルチモーダル対話システム MultiksDial. 電子情報通信学会論文誌, vol.J77-D-II, no.8, pp.1429–1437, August, 1994.

[LPW98] D. J. Litman and S. Pan and M. A. Walker. Evaluating Response Strategies in a Web-Based Spoken Dialogue Agent, In *COLING-ACL 98*, pp.780–786, 1998.

[MT98] 森元 逞, 竹澤 寿幸. マルチモーダル入力, マルチメディア出力の案内システム: MMGS. インタラクション*'98*, pp.177–180, 情報処理学会, 1998.

[NDI98] 中川 聖一, 傅田 明弘, 伊藤 敏彦. マルチモーダル観光案内対話システム 人工知能学会誌, vol.13, no.2, pp.241–251, March, 1998.

[Tak94] 竹林 洋一. 音声自由対話システム TOSBURG II – ユーザ中心のマルチモーダルインタフェースの実現に向けて – 電子情報通信学会論文誌, vol.J77-D-II, no.8, pp.1417–1428, August, 1994.

[TM98] T. Takezawa and T. Morimoto. A Multimodal-Input Multimedia-Output Guidance System: MMGS, In *Fifth International Conference on Spoken Language Processing*, vol.2, pp.285–288, 1998.

[TT99a] 高橋 和子, 竹澤 寿幸. マルチモーダル対話システムにおけるインタラクション機構. インタラクション*'99*, pp.89–96, 情報処理学会, 1999.

[TT99b] 高橋 和子, 竹澤 寿幸. マルチモーダル対話システムにおける対話管理. 情報処理学会第 *58* 回全国大会, vol.4, pp.11–12, 1999.

[WIC94] M. Walker, M. Iida and S. Cote. Japanese Discourse and the Process of Centering, Computational Linguistics, vol.20, no.2, pp.193–231, 1994.

付録

A　評価実験で使用した対話

A.1　例文による対話

場面 1 [光台地区の地図]

1. ATR はどれですか.
2. レストランはありますか.
3. 高の原駅までバスはありますか.
4. 時間はどのくらいかかりますか.
5. 奈良の地図に切り替えて下さい.

場面 2 [奈良の地図]

1. 奈良公園の観光コースを教えて下さい.
2. 時間はどのくらいかかりますか.
3. 大仏は何時から見られますか.
4. いくらかかりますか.
5. 近くで昼食は食べられますか.
6. 奈良公園の中にはないんですか.
7. 大仏殿から歩いて行けますか.
8. これは何ですか [茶店の隣の建物を○で囲む].
9. 奈良のみやげものを教えて下さい.
10. どうもありがとう.

A.2　課題付き自由対話

[課題 1]
　あなたは奈良公園を観光しようと思っています．特に，大仏殿を見たいと思っており，時間がかかるようなら昼食も食べて帰るつもりです．3 時には京都に戻らなければなりません.

[課題 2]
　あなたは奈良公園を観光しようと思っています．特に，二月堂と大仏殿を見たいと思っていますが，場所や行き方など詳しい情報を知りません．土産物を買って帰るつもりです．夕方には京都に戻らなければなりません.

[課題 3]
　あなたは今近鉄奈良駅におり，これから奈良県新公会堂で行なわれる会議に出席するためそちらに行かねばなりません．あなたは行き方についてはまったく知りません．会議が早く終われば奈良公園を歩いて近鉄奈良駅まで帰ろうと思っていますが，遅くなればバスを使おうと思っています.

B　アンケート集計結果

0. プロフィール

1. 年齢と性

年齢	10代	20代	30代	40代	計
男性	2	4	6	0	12
女性	0	3	3	1	7
計	2	7	9	1	19

2. 出身地 (10才までに最も長くいた県名)
大阪府 6 名，神奈川県 2 名，京都府 2 名，
岩手県，千葉県，静岡県，愛知県，兵庫県，鳥取県，岡山県，愛媛県，福岡県各 1 名

3. コンピュータの使用経験
(a) あり − 18 名　(b) なし (ほとんどない) − 1 名

4. 音声入力の経験
(a) あり − 4 名　(b) なし (ほとんどない) − 15 名

1. 音声入出力について

1. 例文による対話で話した言葉は正しく認識されましたか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	8	9	2	0	0

2. 課題付き自由対話で自分の表現したいことを自然に表現できましたか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	0	0	2	10	7

3. マウスによる音声入力操作はやりやすかったですか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	7	6	4	2	0

2. 画像入出力について

1. 画面の動きについて

(a) 速さ

	速い		⟷		遅い
評価	5	4	3	2	1
人数	1	3	11	4	0

(b) 大きさ

	動きすぎ		⟷		もっと動いた方がいい
評価	5	4	3	2	1
人数	0	6	10	3	0

(c) 種類 (回転，ズーム等)

	適当		⟷		わかりにくい
評価	5	4	3	2	1
人数	1	9	3	4	1

無回答:1 名

(d) 画面上の指示は正しく認識されましたか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	13	2	1	0	2

無回答:1 名

3. インタラクションについて

1. 話しやすかったですか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	0	2	4	9	4

2. 課題付き自由対話で自分の目的とする情報は得られましたか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	0	0	1	9	9

3. 応答文の表現 (音声の抑揚ではなく内容) は自然だと思いますか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	5	5	5	2	2

4. 全体

1. 応答時間はどうでしたか?

	短い		⟷		長い
評価	5	4	3	2	1
人数	0	1	6	9	3

2. 画面にうさぎのキャラクタを出していますが, どう感じましたか? (複数回答可)

   (a) 待ち時間が楽しめてよい – 12 名

   (b) 退屈 – 0 名

   (c) キャラクターのバリエーションが必要 – 7 名

   (d) 特に何も感じなかった – 2 名

   (e) その他 – 5 名

   - 個人的には無くてもよい
   - 考えているジェスチャが必要
   - 認識されなかったときの表情がほしい
   - 案内が終わったかどうか確認できてよかった
   - 耳が動くことで声入力の確認ができる

3. このような音声画像インタフェースを持つシステムは, 案内システムとして適当だと思いますか?

	はい		⟷		いいえ
評価	5	4	3	2	1
人数	6	9	4	0	0

4. このような音声画像インタフェースを持つシステム (案内システムに限らない) を使ってみたいと思いますか?

(a) 使ってみたい – 2 名

(b) 以下の条件付で使ってみたい (複数回答可) – 15 名

i. データベースが整備されたら – 9 名

ii. 音認識率があがったら – 14 名

iii. 画像がもっときれいになったら – 3 名

iv. 応答速度がもう少し早くなったら – 7 名

v. ノートパソコンで動いたら – 1 名

vi. その他 – 2 名

- 効率的な対話が可能になれば
- 認識可能なきき方がある程度例示されていた方が気楽にきける

(c) 使いたくない – 2 名
理由:

- 機械に向かって一人で話しかけることにためらいがある
- クレジットカード等，信用問題，金に絡むことには使いたくない

5. その他意見 (主なものの抜粋)

- 音声認識，対話について
  - 音声認識率が低く，ストレスを感じる
  - 「どうやって～」や「～まで行きたいんですが」はよく使用すると思うがもっといろんな質問に対応してほしい
  - 音声認識失敗の原因に応じて「もっとゆっくり」「もう少しはっきり」等，多様な問い返しパターンが使い分けられているとよい
  - 理解・対話状況を見せられないか?
- 画像について
  - 地図上に観光ルート，道順等を表示したほうがよい
  - 建物の名前はあらかじめ表示していてほしい
  - 3D のターンがかえって方向感覚を混乱させてしまっている
  - 地図を前もって見ている人でなければ，画面が速く動きすぎる
- その他
  - 現在の音声認識 + 自然言語処理はどういうものかを知ることができてよかった
  - 画面に向かって話しかけるのには，とても違和感がある