

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

高橋 和子 竹澤 寿幸

〒 619-0288 京都府相楽郡精華町光台 2-2

ATR 音声翻訳通信研究所 第四研究室

(Phone) +81-774-95-1351

(Fax) +81-774-95-1308

(e-mail) { ktaka,takezawa }@itl.atr.co.jp

マルチモーダル対話システムにおける対話管理について述べる。インタ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、省略されたもの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、省略されたもの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、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現発話が表示対象を表す。このデータは発話ごとに内容が更新される。これによって、インタ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。

An Interaction Mechanism for Multimodal Dialogue Systems

Kazuko TAKAHASHI and Toshiyuki TAKEZAWA

ATR Interpreting Telecommunications Research Laboratories

2-2, Hikaridai, Seika-cho, Souraku-gun, Kyoto, 619-0288, JAPAN

(Phone) +81-774-95-1351

(Fax) +81-774-95-1308

(e-mail) { ktaka, takezawa }@itl.atr.co.jp

We describe the dialogue processing for multimodal dialogue systems. We find a lot of ellipsis in multimodal dialogue systems with high interaction. The system cannot search its database if the ellipsis is not resolved. In this paper, we propose the mechanism for ellipsis resolution using the method of history storing/retrieving. In this method, the object indicated by the current utterance is represented as the data that comprises all the necessary terms related to the current utterance, which appeared in the utterances in the past. The data is updated according as a dialogue proceeds. We formalize the method and show that it can enhance the feasibility of the dialogue system and also realize the system with more natural interaction.

1 はじめに

ここ数年、GUI にな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ものになるため、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った特徴が見られる。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。

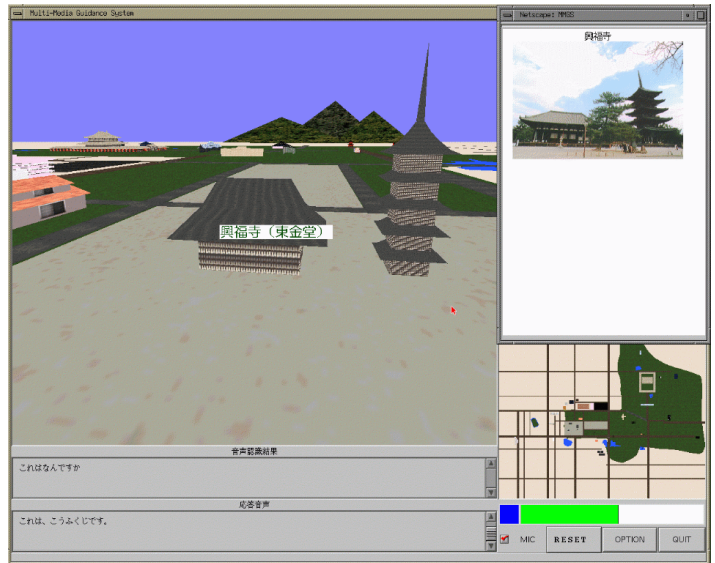


図 1: MMGS の表示例

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。

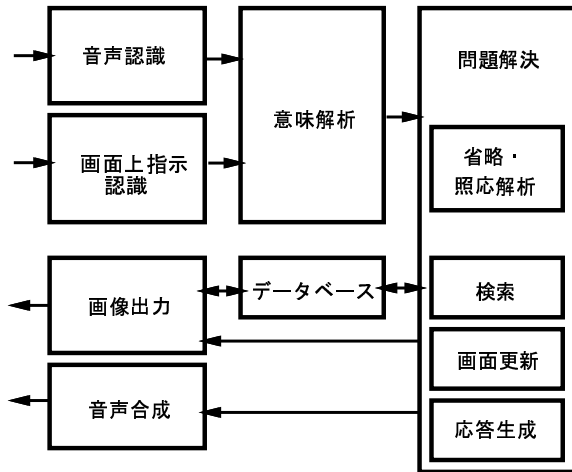


図 2: MMGS のシステム構成

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、最後に5章で結論を述べる。

2 マルチモーダル対話システム

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をもとに Q 話題を生成し、データベースの検索を行なう。その結果と Q 話題から A 話題を生成し、それを基に画面の更新情報や応答文を生成する。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。

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。

2.2 意味表現

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*、 $(transfer \dots)$ 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 インタラクシオン機構

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 $Ap = (label, (fn_1, fv_1), \dots, (fn_k, fv_k))$ 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

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。

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。

[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 $i (i = 1, \dots, k)$ 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 $i (i = 1, \dots, k)$ が存在しない時

データベースに $(label, (fn, fv), (fn_1, fv_1), \dots, (fn_k, fv_k))$ 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 $i (i = 1, \dots, k)$ 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 $i (i = 1, \dots, k)$ 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 $i (i = 1, \dots, k)$ 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 $i (i = 1, \dots, k)$ 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ければ、その名前を lbl とする。複数見つければ、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(「見学時間を教えてください」「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(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(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(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 (discourse 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、語句の意味の近さをかなり細かく定義せざるを得ない。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 おわりに

マルチモーダル対話システムの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。これによって、インタ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。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用可能である
- 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り、これは、今後の実験によって評価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 Interfaces, *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.
- [MT98] 森元 暉, 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 In *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.
- [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.