音声言語情報処理 32－4
（2000．7．14）

# 複数話者の発話が混在する発話系列を用いた対話理解

宮崎 昇　平沢 純一　中野 幹生　相川 清明

NTT コミュニケーション科学基礎研究所
〒 243-0198 神奈川県厚木市森の里若宮 3-1
{nmiya,nakano}@atom.brl.ntt.co.jp,　{jun,aik}@idea.brl.ntt.co.jp

本報告では対話理解に関する理解規則の新しい記述方法について述べる．本手法は話者識別情報を含む任意長の入力発話系列に対して対話理解規則が適用される点に特徴がある．任意長の発話系列に対する理解規則の集合による対話記述は非決定性オートマトンによる対話記述となり，入力発話に対し決定的な状態遷移を行う対話記述に比べ簡素な対話記述が可能となる．また，対話システム自体を対話参加者ととらえシステム発話を入力の一部と見なした場合には、ユーザとシステム間のやり取りに関しても簡素な記述で柔軟な対話理解が行われる．本手法は話者数を一般化して，2名の対話だけでなく3名以上の参加者による多人数対話に対しても対話理解規則の記述が可能である．

キーワード: 音声対話, 対話理解, 話者識別変数, 発話系列

# Dialogue understanding using utterance sequence by multiple speakers.

Noboru Miyazaki, Jun-ichi Hirasawa,
Mikio Nakano, Kiyoaki Aikawa

NTT Communication Science Laboratories
3-1, Morinosato Wakamiya, Atsugi City 243-0198, Japan

This paper presents a new method to understand dialogue. We use dialogue understanding rules that accept an input utterance sequence where length of each sequence may be different. This is a non-determinisitic way to describe system's internal state transition, and results in a more simple description of dialogue understanding rules compared with a determinisitic way. Furthermore, regarding system utterances themselves as a part of inputs allows a more simple description about interaction between the user and the system. Our method has a general structure to handle not only dialogue with two speakers but also multi-party dialogue.

keywords: spoken dialogue, dialogue understanding, speaker identifier, utterance sequence

# 1 はじめに

音声対話システムによるインターフェースを構築する際，ユーザ発話に与える自由度と対話システム記述の簡素さのバランスが一つの大きな問題となる．発話の自由度が下がればユーザの負担が増大するが，発話の自由度を上げるためにタスク依存の複雑なシステム状態遷移を手続き的に記述するアプローチは対話システムのタスクポータビリティを低め，更にシステム開発のコストを増加させる．

そこで本研究ではより簡素なシステム記述で柔軟な対話進行を可能とするような新たな対話理解手法を提案する．

本報告ではまずシステム制御の記述方法に着目し，2節で入力発話に対し非決定的な状態遷移を記述する手法がより柔軟な対話処理を可能とするが，対話システムへ適用した場合にシステムの内部状態変化がユーザに開示されず，応答の一貫性が失われる場合があることを示す．この問題の回避を容易にしつつ，さらに話者間のやり取りに関する柔軟な対話処理を実現するため，3節で複数話者からの発話が混在する発話系列に対する対話理解規則を導入し，4節で対話システムへ適用した例を示す．

# 2 対話制御記述

| | |
|---|---|
| $S_1$ | ご用件をどうぞ |
| $U_1$ | 神奈川県 [香川県] の |
| $U_2$ | 天気を教えてください |
| $S_2$ | 香川県の天気ですね |
| $U_3$ | 神奈川県です |
| $S_3$ | 神奈川県の天気ですね |
| $U_4$ | はい |

図 1: 対話例 1

対話制御記述について考察するため，例として図1の対話を考える．ここで $S_n$ はシステム発話，$U_n$ はユーザ発話を表わす．ユーザ発話内で [] に囲まれた部分は音声認識による認識誤りを想定している．

## 2.1 決定性オートマトン

従来の対話理解法として，ユーザ発話を入力とする決定性オートマトンを用いるものがある [3, 4, 6]．ここでは図1の対話例を，図2の決定性オートマトンで処理する場合を考える．図2の太い破線矢印は状態遷移後のシステム発話であり，状態間を結ぶアーク上の発話はユーザからの入力発話である．システムは S00 でシステム発話 $S_1$ を出力し，$U_1U_2$ により S02 を経由して S04 へ遷移し $S_2$ を発話する．こ

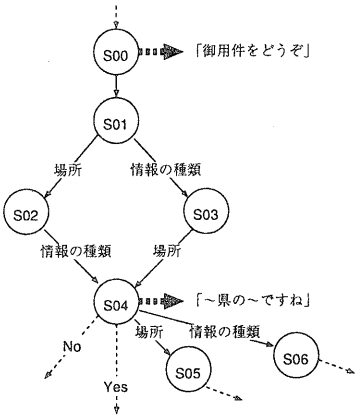


図 2: 決定性状態遷移による対話記述

こでユーザからの訂正入力 $U_3$ を理解し応答 $S_3$ を出力するには，S04 から情報の種類や場所それぞれについてのアークと遷移先 S05，S06 が定義されている必要がある．このように，自由度の高いユーザ発話の処理には想定される全ての状態遷移を明示的に記述する必要があり，記述コストが高い．

## 2.2 非決定性オートマトン

これに対し，中野ら [2] の多重文脈法はユーザ入力を非決定性のオートマトンで処理し，柔軟な対話理解を実現する．例えば図3のオートマトンを用いると，図1 $U_2$ までの対話は

(経路 1) S00 → S01 → 神奈川 [香川] 県の天気を教えてください → S04 → 神奈川県です → S01
(経路 2) S00 → S01 → 神奈川 [香川] 県の → S01 → 天気を教えてください → S03 → 神奈川県です → S04
(経路 n) ...

と複数の解釈が成立し，経路に関する適切な重みづけにより経路2が選択されれば $S_3$ が出力され，対話が進行する．

しかし非決定性オートマトンを用いると，不適切な経路選択によりシステム出力の一貫性が失われる場合がある．例えば図1の対話の後，さらに日付を聞くシステムを想定し図3のオートマトンを用いると，対話例2(図4) が成立する場合がある．これは以下の二つの経路 A，B の優先度が対話中に変化する場合に発生する。対話システムは一度に1種類の発話しか出力できないため、ある時刻において最も優先度の高い経路の状態に応じて出力発話が決定する．

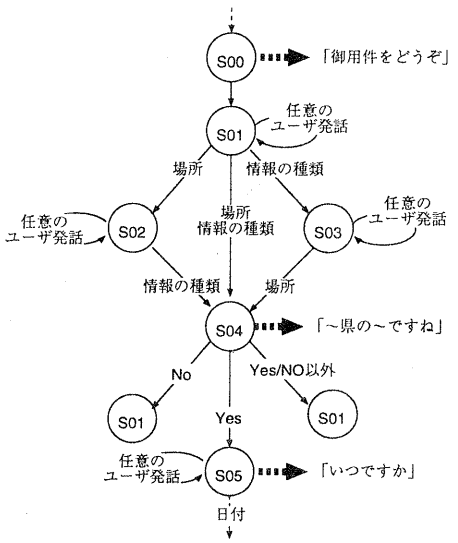


図 3: 非決定性状態遷移による対話記述

| | |
|---|---|
| $S_1$ | ご用件をどうぞ |
| $U_1$ | 神奈川県の天気を |
| $U_2$ | 教えてください |
| $S_2$ | 神奈川県の天気ですね |
| $U_3$ | はい |
| $S_3$ | いつですか? |
| $U_4$ | 今日です [京都です] |
| $S_4$ | 京都の天気予報ですね |
| $U_5$ | 違います |
| $S_5$ | いつですか? |

図 4: 対話例 2

これを表すため経路 A，B で斜体で示される状態は優先度が低い状態であるとする。対話例図 4 において，$S_3$ までは経路 A に添った対話であったのが，ある理由 ($U_4$ の音響スコアが高い，等) で $U_4$ の直後に経路 B の優先度が上がり S04 で $S_4$ を出力し，さらに $U_5$ の後，経路 A の優先度が上がった場合に $S_5$ が出力される．

(経路 A) S00 → S01 → 神奈川県の　天気を教えてください → S04 → はい → S05 → 今日 [京都] です → *S05* → 違います → S05
(経路 B) *S00* → *S01* → 神奈川県の → *S01* → 天気を教えてください → *S03* → はい → *S03* → 今日 [京都] です → S04 → 違います → *S01*

一般に，$S_5$ が出力された時点で，システムの場所に関する内部状態が神奈川であるか京都であるか，あるいは何も値が入っていないかをユーザが把握することは困難である．これはシステム内部で経路 A および経路 B の二種類の文脈が混在し，ユーザに各時点での文脈変更が開示されないためである．

### 2.3 システム発話を入力とするオートマトン

本研究は複数の話者を入力情報源とし，話者識別変数を含む発話の系列を入力として対話を理解する手法を提案する．さらにその特別な場合として対話システムの出力を情報源の一つとし，システム出力を自己回帰的に入力の一部とする．これによりシステムの内部状態とシステム発話との間の不整合が生じにくくなる．対話例 図 4 では，ユーザ発話を入力とする場合，S03 から $U_3U_4$ で S04 へ遷移するパスが発話 $S_5$ の原因であったが，$U_3$ に対する S03 の自己遷移は音声認識の沸き出し誤りを考慮すれば必ずしも不合理な遷移とは考えられない．これに対しシステム発話を入力に加える場合，S03 から 発話列 $S_2U_3S_3U_4$ で S04 へ遷移するパスをシステム構築者が意図的に許さなければ発話 $S_4$，$S_5$ は発生しないため，既に発話されたシステム発話と矛盾するパスが生じにくくなると考えられる．

## 3 発話系列を用いた対話理解

### 3.1 システム記述

音声対話システムを含む対話参加者の集合を $\mathcal{Z}$ とし，発話やフレーズなど，ある処理単位で区切った入力の集合を $\mathcal{W}$ とする．時刻 $t$ における入力 $x_t$ を以下の形式で表現する．

$$x_t \quad = \quad (w_t, z_t) \tag{1}$$

ここで，$w_t \in \mathcal{W}$，$z_t \in \mathcal{Z}$ である．また，$x$ の時系列を $\mathbf{X}_t = \{x_1, x_2, \ldots, x_t\}$ とする．

対話理解システム内部での理解の状態は有限の集合 $\mathcal{S}$ で表現し，時刻 $t$ における対話理解システムの内部状態 $s_t \in \mathcal{S}$ の時系列を $\mathbf{S}_t = \{s_0, s_1, \ldots, s_t\}$ とする．$s_0$ はシステムの初期状態である．時刻 $t$ は 1 から開始し，$x$ が入力される度に 1 づつ増加する．

対話理解システムは $t = 1$ における初期状態

$$\mathbf{S}_1 = \{s_0\}, \qquad \mathbf{X}_1 = \{\phi\} \tag{2}$$

から動作を開始し，時刻 $t$ において $x_t$ が発生する度に対話理解規則の集合 $\mathcal{R}$ および関数 $f$，$g$ を用いて図 5 の処理を行う．ここで $\Omega_t$ は時刻 $t$ において適

1. $\mathbf{X}_t = \{\mathbf{X}_{t-1}, x_t\}$
2. $\Omega_t = f(\mathcal{R}, \mathbf{X}_t, \mathbf{S}_{t-1})$
3. $\Omega_t \neq \phi$ ならば $r_t = g(\Omega_t)$ を $\mathbf{X}_t$ および $\mathbf{S}_{t-1}$ に適用し $s_t$ を得る.
4. $\Omega_t = \phi$ ならば $s_t = s_{t-1}$
5. $\mathbf{S}_t = \{\mathbf{S}_{t-1}, s_t\}$
6. $t = t + 1$

図 5: 対話理解の手続き

用可能な理解規則の集合であり，$g$ は規則の集合 $\Omega_t$ から $r_t$ を一つ選択する関数である．$r_t \in \Omega_t \in \mathcal{R}$ は時刻 $t$ において適用される対話理解規則である．

## 3.2 対話理解規則

対話理解規則 $r \in \mathcal{R}$ は以下の形式で記述される．

$$r = (p, e, \mathbf{U}) \quad (3)$$

$p, e \in \mathcal{S}$ であり，$p$ は $r$ が適用される前提となる理解状態，$e$ は $r$ を適用した結果得られる新たな理解状態である．$\mathbf{U}$ は $r$ が指定する発話列パタンで，形式的には発話 $x$ の系列として与えられる．

$$\mathbf{U} = (u_1, u_2, u_3, ..., u_m) \quad (4)$$
$$u_i = (v_i, y_i) \quad (5)$$
$$v_i \in \mathcal{W}, \quad y_i \in \mathcal{Z}$$

時刻 $t$ において発話 $x_t$ が得られると，規則 $r$ は式 (6)，(7) が満たされる場合に理解状態 $s_t$ を $e$ とする．

$$p = s_{t-m} \quad (6)$$
$$\mathbf{U} = (x_{t-m+1}, x_{t-m+2}, ..., x_t) \quad (7)$$

規則適用の例を図 6 に示す．ここで，破線で示される発話及び状態は理解規則 $r$ の適用に影響を与えない．

## 3.3 規則の選択

図 5 手続き 2. の関数 $f$ は $\mathbf{X}_t$，$\mathbf{S}_{t-1}$ を入力として式 (6)，(7) を満たす規則 $r$ の集合 $\Omega_t$ を計算する．ここで $\Omega_t$ に複数の規則が含まれる場合，適用すべき規則 $r_t$ は関数 $g$ により決定する．$g$ は規則集合 $\mathcal{R}$ と共にタスクに依存となる．

$\Omega_t$ が複数の規則を含むような，同時刻に複数の規則が適用可能となる対話例を図 7 に示す．この例では以下の二通りの解釈が可能となる．

(解釈 1)

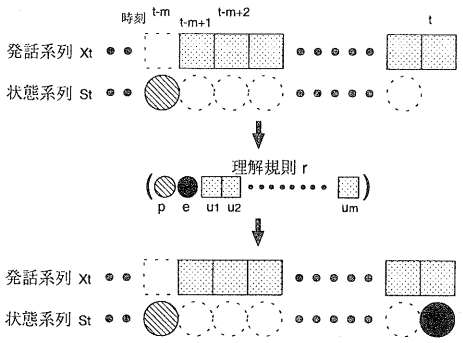


図 6: 理解規則の適用

| 時刻 | | 発話 |
|---|---|---|
| 1 | $S_1$ | 神奈川の |
| 2 | $S_2$ | 天気についてですね |
| 3 | $U_1$ | 違います |
| 4 | $U_2$ | 気温です |

図 7: 対話例 3

1. 発話列 $S_1S_2U_1$ により場所及び情報の種類に関して理解状態をクリア
2. $U_2$ を単独発話として解釈し，情報の種類は「気温」であるとする

(解釈 2)

1. 発話列 $S_1S_2U_1U_2$ により「神奈川の気温」をユーザの要求とする

解釈 1 および解釈 2 をそれぞれ実行する規則が $\mathcal{R}$ に含まれる場合，$\Omega_4$ には $U_2$ を単独発話として解釈する規則と発話系列 $S_1S_2U_1U_2$ に対して「神奈川の気温」をユーザの要求と解釈する規則が含まれ，関数 $g$ が必要となる．

## 3.4 対話システムへの応用

$\mathcal{Z} = \{$ ユーザ, システム $\}$ の条件下で本手法を対話システムの対話理解部として用いたシステム構成を図 8 に示す．対話理解規則はデータとして与えられ，対話理解部は手続き図 5 を行う．発話生成部は対話理解部から状態系列を参照し，必要な発話を生成する．

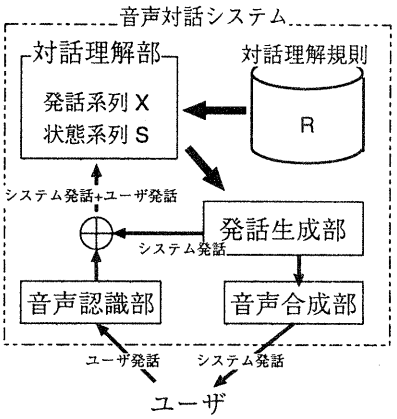


図 8: 対話システム構成

# 4 システム実装例

## 4.1 タスク定義

本手法を用いて実装した対話システムの例を示す．タスクは気象案内システムを想定し，ユーザがシステムに対して「場所」「日付」「情報の種類」の三種類を指定し，システムがそれらの全てについて確認を終えることをタスクとする．「情報の種類」とは天気，気温，湿度，台風情報などの気象情報の区分を指す．システム内部状態 $s$ はフレームにより記述する [1]．フレームは場所，日付，情報の種類の値 $(place, date, category)$ と，それぞれの確認フラグ $(pf, df, cf)$ の 6 個のスロットからなる．確認フラグは $\mathbf{T}$，$\mathbf{F}$ いずれかの値をとり，$\mathbf{F}$ は未確認，$\mathbf{T}$ は確認済の状態を示す．

$$s = (place, pf, date, df, category, cf) \quad (8)$$

$$s_0 = (null, \mathbf{F}, null, \mathbf{F}, null, \mathbf{F}) \quad (9)$$

発話 $w$ は記述の簡素化のため，$\mathcal{W}_c$(単語カテゴリ)，$\mathcal{W}_m$(意味)，$\mathcal{W}_a$(モダリティ) を用いた素性のセットで表現する．すなわち

$$w = (w_c, w_m, w_a) \in \mathcal{W} \quad (10)$$

$$w_c \in \mathcal{W}_c, \quad w_m \in \mathcal{W}_m, \quad w_a \in \mathcal{W}_a \quad (11)$$

各素性値の値を図 10 に示す．本手法では個別の発話でなく発話時系列を解釈の対象とするため，かならずしも個別発話の意図を正確に決定する必要性がない．そのためモダリティが談話タグ [5] より低い抽象度で与えられている．「だ」というモダリティは，「〜です」「〜だ」などの語尾を持つ，という程度の意味を持つ．また，$\Omega_t$ から適用規則を一つに決定する関数 $g$ は，簡単のため $\Omega_t$ のうち最も $\mathbf{U}$ の長い規則を選択するものとした．これは断片的な発話解釈の積み重ねに比べて，長い文脈を考慮した規則はより正確な解釈を与えるというヒューリスティックスに基づいている．

```
規則 1
p1      ( *1  F  *3  *4  *5  *6)
e1      ( *x  F  *3  *4  *5  *6)
U1      ( ( 場所 *x だ Usr ) )

規則 2
p2      ( *1  *2  *3  F  *5  *6)
e2      ( *1  *2  *x  F  *5  *6)
U2      ( ( 日付 *x だ Usr ) )

規則 3
p3      ( *1  *2  *3  *4  *5  F)
e3      ( *1  *2  *3  *4  *x  F)
U3      ( ( 情報の種類 *x だ Usr ) )

規則 4
p4      ( *x   *  *3  *4  *5  *6)
e4      ( null F  *3  *4  *5  *6)
U4      ( ( 場所       *x    -  * )
          ( 対話進行   打消   -  Usr ) )

規則 5
p5      ( *x  F  *y  F  *z  F)
e5      ( *x  T  *y  T  *z  T)
U5      ( ( 日付       *y   -  Sys )
          ( 場所       *x   -  Sys )
          ( 情報の種類 *z   か Sys )
          ( 対話進行   肯定 -  Usr ) )

規則 6
p6      ( *x   F *y   F *z   F)
e6      ( null F null F null F)
U6      ( ( 日付       *y   -  Sys )
          ( 場所       *x   -  Sys )
          ( 情報の種類 *z   か Sys )
          ( 対話進行   否定 - Usr ) )

規則 7
p7      ( *x   F  *y  F  *z  F)
e7      ( *x1  F  *y  F  *z  F )
U7      ( ( 日付       *y    -  Sys )
          ( 場所       *x    -  Sys )
          ( 情報の種類 *z    か Sys )
          ( 対話進行   否定  -  Usr )
          ( 場所       *x1   -  Usr ) )

規則 n  ...
```

図 9: 理解規則例

## 4.2 規則集合

対話理解規則の一部を図 9 に示す．規則は式 (3) の形式で直接記述されるが，記述の簡素化のため発話は素性のセットに対する制約で記述し，規則適用はマッチングで実現する．規則内のシンボルのうち '*' ではじまるものは変数であり，同一規則内では同一変数の値は共通である．ただし '*' のみで記述されるシンボルは規則適用に影響を与えない．例えば規則 4 は，場所に関する理解状態が '*x' である場合にいずれかの話者が ' *x' を発話し，直後にユーザが打消を意味する発話 (例えば，「じゃない」など) を

| | |
|---|---|
| $\mathcal{W}_c$ | 場所, 日付, 情報の種類, 対話進行 |
| $\mathcal{W}_m$ | 東京, 千葉,..., 今日, 明日,..., 気温, 湿度,... |
| $\mathcal{W}_a$ | だ, ですか, ね |

図 10: 素性値

行った場合に，場所に関する理解状態を null，場所に関する確認フラグを未確認とする規則である．今回実装された理解規則の総数は 42 である．

### 4.3 発話生成

システム発話は最新の理解状態 $s_t$ に対して，1) 値が null のスロットは質問する．2) 値が入っているが未確認のスロットは確認する．3) 確認の際，可能ならばまとめて確認する，等の規則により生成する．

### 4.4 動作例及び効果

図 11 に本手法による対話システムの動作例を示す．確認対話 $S_1S_2U_4$ が理解状態に反映されるため $U_5$ は理解状態に影響を与えず，結果的に図 4 $S_5$ のような発話を避けられる．

また，図 12 時刻 16 では $S_7...U_7$ が規則 6 で解釈されるが，直後の 時刻 17 では $U_8$ の規則 1 による解釈と $S_7...U_8$ の規則 7 による解釈が成立し，$\Omega_{17} = \{$規則 1, 規則 7$\}$ となる．ここで $g$ により $\mathbf{U}$ の長い規則が選択され $r_{17} =$ 規則 7 となる．このとき，規則 7 は内部状態 $s_7$ と発話系列 $S_7...U_8$ に対して適用され，直前の状態 $s_{16}$ は適用に影響しない．本手法ではこのような非決定的な規則適用が可能であり，対話理解規則の柔軟な記述が可能である．また，全ての時点であらゆる規則の適用判定が行われるため，時刻 20 からシステムが「明日の神奈川県の湿度ですか」と確認を行う際にユーザ発話 $U_{10}$ が割り込んだ場合も，規則 4 により対処可能である．

これらの対話理解は全て発話生成部と独立に行われる．このため対話制御，ユーザ発話理解，システム発話生成の関係を決定性オートマトンやスクリプト言語を用いて記述する手法と比べ，柔軟な対話理解が可能となる．

## 5 まとめ

入力発話に話者識別変数を含み，発話系列に対して対話を理解する手法を提案した．また対話システムを話者と見なした場合，システム発話を入力の一部とした対話理解規則を記述でき，簡素な対話理解記述で柔軟な対話理解が行われることを示した．本手法の対話理解規則は 3 人以上の多人数対話に対しても枠組みを変更せずに適用可能である．今後は音声を用いた音声対話システムに適用した後，規則に含まれるシステム発話を積極的に利用したシステム発話生成手法の検討を行う．

| 時刻 | 話者 | 発話 | 内部状態 |
|---|---|---|---|
| 0 | | | ( null F null F null F ) |
| 1 | $U_1$ | 神奈川の | ( 神奈川 F null F null F ) |
| 2 | $U_2$ | 天気を | ( 神奈川 F null F 天気 F ) |
| 3 | $U_3$ | 教えて下さい | ( 神奈川 F null F 天気 F ) |
| 4 | $S_1$ | 神奈川の | ( 神奈川 F null F 天気 F ) |
| 5 | $S_2$ | 天気ですか | ( 神奈川 F null F 天気 F ) |
| 6 | $U_4$ | はい | ( 神奈川 T null F 天気 T ) |
| 7 | $S_3$ | いつですか | ( 神奈川 T null F 天気 T ) |
| 8 | $U_5$ | [京都です] | ( 神奈川 T null F 天気 T ) |
| 9 | $S_4$ | いつですか | ( 神奈川 T null F 天気 T ) |
| 10 | $U_6$ | 今日です | ( 神奈川 T 今日 F 天気 T ) |
| 11 | $S_5$ | 今日ですか | ( 神奈川 T 今日 F 天気 T ) |
| 12 | $U_7$ | はい | ( 神奈川 T 今日 T 天気 T ) |

図 11: 動作例 1

| 時刻 | 話者 | 発話 | 内部状態 |
|---|---|---|---|
| 9 | $S_4$ | 今日の | ( 東京 F 今日 F 気温 F ) |
| 10 | $S_5$ | 東京の | ( 東京 F 今日 F 気温 F ) |
| 11 | $S_6$ | 気温ですか | ( 東京 F 今日 F 気温 F ) |
| 12 | $U_6$ | 湿度です | ( 東京 F 今日 F 湿度 F ) |
| 13 | $S_7$ | 今日の | ( 東京 F 今日 F 湿度 F ) |
| 14 | $S_8$ | 東京の | ( 東京 F 今日 F 湿度 F ) |
| 15 | $S_9$ | 湿度ですか | ( 東京 F 今日 F 湿度 F ) |
| 16 | $U_7$ | いいえ | ( null F null F null F ) |
| 17 | $U_8$ | 神奈川です | ( 神奈川 F 今日 F 湿度 F ) |
| 18 | $S_{10}$ | 今日の | ( 神奈川 F 今日 F 湿度 F ) |
| 19 | $U_9$ | 明日 | ( 神奈川 F 明日 F 湿度 F ) |
| 20 | $S_{11}$ | 明日の | ( 神奈川 F 明日 F 湿度 F ) |
| 21 | $S_{12}$ | 神奈川の | ( 神奈川 F 明日 F 湿度 F ) |
| 22 | $U_{10}$ | じゃなくて | ( null F 明日 F 湿度 F ) |
| 23 | $S_{13}$ | どこですか | ( null F 明日 F 湿度 F ) |

図 12: 動作例 2

## 謝辞

日頃より御指導頂く NTT コミュニケーション科学基礎研究所メディア情報研究部萩田紀博部長に感謝致します．また，貴重なコメントを下さった同研究部マルチモーダル対話研究グループの皆様に感謝致します．

## 参考文献

[1] D.Goddeau et al. A form-based dialogu manager for spoken language appilcations. In *Proc.* **ICSLP-95**, pp. 701–704, 1995.

[2] Mikio Nakano, Noboru Miyazaki, Jun-ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. "Understanding Unsegmented User Utterances in Real-Time Spoken Dialogue Systems". In *Proc.***ACL99**, pp. 200–207, 1999.

[3] 青山一美, 平野泉, 菊地英明, 坪川拓史, 白井克彦. "音声対話システム汎用プラットフォームの検討". 情処研報, Vol. 2000-SLP-30-2, pp. 7–12, 2000-2.

[4] 麻生英樹, ジョンフライ, 浅野太, 速水悟, 伊藤克亘, 本村陽一, 栗田多喜夫, 松井俊浩. "事情通ロボットの音声対話インターフェース". 情処研報, Vol. 2000-SLP-30-3, pp. 13–20, 2000-2.

[5] 市川他. "談話タグ標準化の現状". 人工知能研資, Vol. SIG-SLUD9703-7, pp. 41–48, 1997-3.

[6] 桐山伸也, 広瀬啓吉. "文献検索タスク音声対話システムの評価実験". 平十二春音響学会講演論文集, Vol. 3-8-7, pp. 79–80, 2000.