自然言語を理解するソフトウエアロボット：傀儡

新山祐介[†] 徳永健伸[†] 田中穂積[†]

我々は自然言語を理解する仮想世界上のロボットを，ユーザの音声によって対話的に動作させるシステム傀儡（かいらい）を開発している．本論文では，まずこれまでの自然言語による対話システムの研究を概観する．次に我々のシステム傀儡の意義と機能を述べ，これを実装するにあたって生じるいくつかの問題およびその解決手法を述べる．本研究では仮想世界上のロボットを動作させるために，これまでの対話システムで扱われてきた問題（名詞句と照応の解決，発話行為の解釈など）のほかに，これまでほとんど扱われてこなかった問題（視点を考慮した位置関係の解釈，動作に関する漠然性の問題など）を解決する必要がある．これらの問題を解決するため，傀儡では文脈やユーザの視点，仮想世界の状況などを考慮に入れる．最後に今後の研究課題について述べる．

"Kairai" — Software Robots Understanding Natural Language

Yusuke Shinyama,[†] Takenobu Tokunaga[†] and Hozumi Tanaka[†]

We are developing a system named Kairai, in which virtual robots understand natural language instructions and act on them in the virtual world. In this paper, first we review the existing dialogue systems. In these systems, problems such as resolution of anaphora and ellipsis, interpretation of speech act has been tackled. In addition to these problems, our research project tackles problems such as interpretation of spatial expression with respect to user's viewpoints, vagueness in instructions, and so forth. Next we describe these problems in realizing the system, and the solutions. To solve problems, our system deals with the context of conversation, the user's viewpoint, the situation of virtual world and so on. Finally, we conclude the paper and mention future work.

1. はじめに

我々は自然言語を理解する仮想世界上のロボットを，ユーザの音声によって対話的に動作させるシステム傀儡（かいらい）を開発している．言語によってロボットに世界を操作させる対話システムとしては，Winograd による SHRDLU が先駆的である[11]．SHRDLU では，ユーザは英語でシステムに積み木を動かすよう指示する．システムはユーザの入力した文を解析し，積み木を動かすための手順を自動的にプランニングし，曖昧な点があればそれをユーザに問い返す．

我々が開発するシステム傀儡は，SHRDLU とは以下の点で異なっている．近年の CG 技術の発達により，現在はより複雑な動作が計算機上で表現できるようになっている．そこで我々はコンピュータグラフィックス，音声認識，および自然言語処理を密接に関連づけることによって，SHRDLU よりも複雑な動作が可能で，扱える言語表現の複雑さも増したシステムの開発を目指している☆．

また，我々は話し言葉の理解に重点を置いている．ユーザは実世界に近い仮想世界を目にして発話するため，その発話には従来の言語理解システムでは扱われてこなかったさまざまな現象が現れる．たとえば「それはもっと前だ」「そうじゃない」などの行動に直結した発話行為や，「ちょっと」「かなり」などの程度を表す語句にみられる漠然性の問題，「あなたの右側」などにみられるような視点によって変化する位置関係などである．CG と音声認識を対話システムと結合することによって，これらの問題を扱う新しい研究分野が生まれる．

一方，対話システムにおける照応や発話行為の解釈に関しては過去多くの研究がなされているが，いまだ解決されていない問題も多い．たとえば SHRDLU で

† 東京工業大学情報理工学研究科計算工学専攻
Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

☆ 我々のシステムは現在 web 上でフリーで配布しており，誰でも PC 上で動作させることができる．参照：http://tanaka-www.cs.titech.ac.jp/kairai/

は，システムは文中の名詞句をすべて実際の仮想世界を探索することによって決定している．しかしこれはユーザの視点の変化を考慮しておらず，我々が日常的に目にする状況とは異なる．発話行為の解釈に関しては Allen らによる TRAIN システムが有名であるが[1]，これはデータベースの問合せを行うシステムであり，我々のようにロボットが仮想世界を操作するような状況では，観察される発話行為は異なる．ほかにも言語表現から 3 次元空間を構成する試みがあるが[5),10),14)]，これらはいずれもユーザによる視点の変化や漠然性の問題を考慮しておらず，扱っている言葉もおもに書き言葉が中心である．

このようなシステムを構築するにあたっては，物体や空間的な情報を表す言語表現と，行為を表す言語表現についての詳しい研究が不可欠である．たとえば我々が対象とするシステムでは，ユーザは「そこにあるものを，向こうへ押せ」などといった曖昧な表現をすることが多く，こういった文の意味はそれを解釈する状況によって左右される．実際には我々は字句的な情報だけでなく，周囲のさまざまな状況を考慮して文の意味を決定している．たとえばこれまでの対話からの文脈や，相手との立場関係，自分の身体と物体の大きさ，相手の声調，そして自分の視界や気分といったものである．哲学ではこれらの状況における曖昧性を，意味の両義性を表す曖昧性とは区別して漠然性と呼んでいる．従来の自然言語処理においては，漠然性に関する研究はほとんど行われてこなかった．しかし，我々が想定している状況では，システムは実際にユーザの指示を実行するために，発話の中に含まれるこのような漠然性を解消しなければならない．

2. 自然言語理解システム：傀儡

我々が開発しているシステムでは，ユーザは計算機と共同で仮想世界における物体の配置を行うことができる．仮想世界上には，自然言語を理解するソフトウエアロボットとともに，いくつかの球があらかじめ置かれている．ユーザはロボットに球を指定の場所まで動かすよう，音声によって指示を与える．その結果はアニメーションとしてユーザに提示される（**図 1**）．たとえば本システムでは以下のような指令を与えることができる：

(1) 「馬はその球を押して」
(2) 「もうすこし」
(3) 「ニワトリは右の赤い球の後に行って」
(4) 「もうちょっとその球を右に」

現在のところ，ソフトウエアロボットが可能な動作

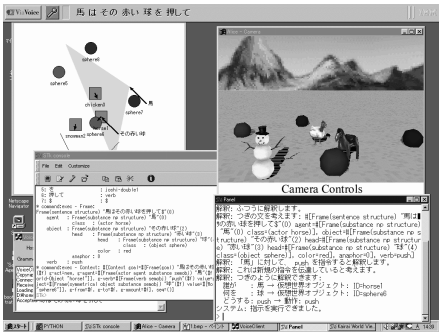

図 1 傀儡の実行画面
Fig. 1 Kairai screenshot.

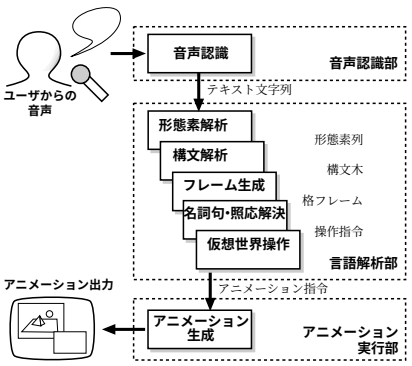

図 2 システムの構成
Fig. 2 System components.

は次の 3 つである：

- 行く（仮想世界上の特定の場所に移動する）
- 向く（特定の方向を向く）
- 押す（物体を指定した距離だけ，あるいは指定した場所まで押す）

各ロボットはユーザの自然言語による指示を理解し，それぞれ個別に行動する．仮想世界上にはロボットのほかにカメラが置かれており，ユーザはこのカメラを通して仮想世界を観察することができる．またこのカメラもロボットの一種であり，ユーザはカメラに対しても指示を与えることができる．ユーザは各ロボットおよびカメラに指示を与えながら逐次的に作業を進める．

図 2 は本システムの構成を示している．図中の点線による枠はおおまかな各コンポーネントを表しており，上からそれぞれ順に音声認識部，言語解析部，そしてアニメーション実行部となっている．内部の長方形はさらに細かなモジュールを表している．

システムの処理は以下のように進む．まずユーザの発話は音声認識部によって文字列に変換され，言語解析部によってアニメーション実行指令に変換される．

これがアニメーション実行部に送られ，ユーザは結果を得る．言語解析部における処理の流れをおおまかに示すと以下のようになる：

(1) ユーザの発話した文を解析し，中間的な意味表現を生成する．
(2) 意味表現からユーザの意図を推測する．
(3) ユーザの意図に基づいて意味表現の中から曖昧な部分を決定し，指令を実行するための手続きを生成する．
(4) 生成した手続きを実行し，仮想世界の状態を変更する．
(5) 仮想世界の状態の変化をアニメーション実行指令として出力する．

ユーザが入力した文はまず形態素解析モジュールに送られる．形態素解析モジュールは単語辞書を用いてこの文を形態素列に変換し，構文解析モジュールに送る．構文解析モジュールは与えられた文法に基づき，この形態素列に対して構文解析を行う．そしてこの結果生成された構文木をフレーム生成モジュールに送る．フレーム生成モジュールはこれをさらに格フレームに変換する．

フレームとは，いくつかのスロットを持つデータ構造である．各スロットは値を持ち，スロットの中にさらに別のフレームを入れ子状に格納することができる．本システムではこの構造を，自然言語をアニメーション指令に変換する中間言語として用いる．フレームのスロットの内容として文の格情報を格納したものを特に格フレームと呼ぶ[4]．たとえば「馬はその赤い球をすこし押して」は，以下のような格フレームに変換される：

```
Frame“ 馬はその赤い球をすこし押して ”
  agent:  馬
  object: その赤い球
  amount: すこし
  verb:   押す
```

次に格フレームは意味・照応解決モジュールに送られる．このような格フレームから実際のアニメーション指令を生成するには，格フレーム中の「馬」「その赤い球」などの名詞句を，実際の仮想世界上のオブジェクトに対応させる必要がある．このため意味・照応解決モジュールは格スロットの名詞句に合致するオブジェクトを仮想世界から探索し，格フレームを修正する．最終的に得られるフレームは次のようなものになる：

```
Frame“ 馬はその赤い球をすこし押して ”
  agent:  actor3
  object: sphere4
  amount: 2
  verb:   push
```

仮想世界操作モジュールはこの情報をもとに次のようなアニメーション生成手続きを作成する：

```
push(actor3, sphere4, 2)
```

3. 解決すべき問題

前章で述べたようなシステムを計算機上に構築するにあたって，解決すべき問題としては次のようなものがあげられる：

発話行為の理解

- 自然言語を用いてユーザが指令をする表現にはさまざまなものがありうる．たとえばユーザは移動を指示するのに「〜に行け」または「〜に行ってください」という表現を使うかもしれないし，あるいは「〜に行ってもらえませんでしょうか」「〜に行けますか」という表現を使うかもしれない．最後の 2 つの文は質問文であるが，これは命令を意味している．また「それはもっと右だよ」「そんなに右じゃない」などの叙述文も命令と見なせる．さらに本システムが想定している話し言葉では，「右」「もうちょい」「違う」などの語だけでも十分に意味が通じることもある．システムはこのような文を適切に解釈するために，ユーザの意図を推測し，ユーザの発話行為を理解する必要がある[2),3)]．

物体や位置，方向の指定

- 前章であげた「押せ」「行け」などの指令は，最終的に計算機に理解可能な手続きに変換される．この手続きに渡されるパラメータは，最終的に仮想世界上のある特定のオブジェクトや点，および領域へのポインタになっている必要がある．仮想世界上のオブジェクトをなるべく簡単に指定できるようにするため，システムは多様な表現を受けつけなければならない．たとえば球を指示するのに「あそこの赤いの」「その右ななめ前あたりにある球」「2 番目に遠い球で，そんなに右じゃないやつ」といった表現が使えることが望ましい．このような表現を処理するためには，システムは言語によって表されたさまざまな制約を理解し，仮想世界を探索する必要がある．また，ユーザの指令によっては，システムはカメラが映している映像を認識する必要もある．たとえば「それら全体

が映るようにカメラをパンせよ」や「ニワトリのトサカを見せて」などといった場合である．

- ユーザは仮想世界の映像を見ながら発話するため，ユーザが発した「あそこ」や「右」などの表現を解釈する際には，システムはユーザの視点や仮想世界の状況を考慮に入れなければならない．また対象となる物体が向きを持っている場合，たとえば「列車の右側」は発話者の向きに関係なく定まるのに対して「テーブルの右のドア」は発話者がどの方向からテーブルを見ているかによって異なる[12]．しかし，ユーザが指令する相手の視点に立って物事を記述する場合もある．たとえば「それ取って」という発話における代名詞「それ」は，相手の目の前に置かれている物体を指している可能性がある[13]．
- ユーザは過去の発話で言及したものを，「それ」「さっきの場所」などの代名詞や連体詞をもちいて言及するかもしれない．システムがこのような照応表現を適切に解釈するためには，ユーザの発話履歴を記録しておき，現在の文脈からなにが適切なのかを探索する必要がある．またユーザは何体かのロボットに別々に指示を与えることができるため，「それ」が指す物体が必ずしも単純に直前に言及されたものであるとは限らない．
- 人間の空間的な位置関係の解釈は，参照物の形状に左右される．たとえば「コップの中の球」がコップに内包されている球を意味するのに対し，「お皿の中の球」は，皿の上に乗せられている球を意味する．また本システムにおける仮想世界は連続な空間であり，「遠い」「大きい」「もっと右」などの程度を表す語句は，実際に指令を実行するときにはある具体的な値を持たなければならない．しかしユーザはこの値を特に指定していないため，ここには漠然性が存在する．一般に，このような値は対象物によってその度合いが変化する[8]．システムは特に指定されないかぎり，人間にとって最も自然な度合いをデフォルトとして使用すべきである．
- すべてを自然言語で指示できるといっても，グラフィカルなインタフェースのほうが依然として望ましい場合もある．たとえばユーザがある位置を指し示すとき，ユーザはマウスなどを用いて，実際に画面上のある位置を指しながら「ここへ…」などと発話できることが望ましい．さらにこのような指令を実行する場合，システムはユーザの音声と動作を同時に取得できる構造になっている必要がある．

漸進的な意味解釈

- ユーザが実際に画面を見ながら指令する際には，ユーザは「あの，その球をもっと右…いや違った，もっと前，いやそんなじゃなくて，もっとそっちのほう，遠く」などといった発話をすることもありうる．このような指示を処理する場合，システムはユーザの発話の終了を待たずに実行を開始し，ユーザの目的を漸進的に推測していかなければならない．また，ユーザは指示の途中で「あっ，そうじゃない」と言うなど，システム側に割込みをかける可能性もある．この場合，システムはユーザの音声にリアルタイムに反応する必要がある．

システムからのフィードバック

- ユーザは複数の解釈を許すような，本質的に曖昧な文を発話してしまうこともある．この場合，このような曖昧性を含んだ発話がなされた場合，システムはそれを無理矢理特定の解釈にはめこんでしまうことはすべきでなく，ユーザに曖昧な部分を問い返すべきである．
- ユーザはシステムが理解できない複雑な指令を発話することもある．この場合，システムはユーザの発話の中から理解できる部分を探し「これはできるが，これはできない」といったガイドを表示することが望ましい．

その他

- システムはユーザになるべく現実世界の物理法則に従ったアニメーションを表示することが望ましい．そのためロボットは，物体間の相互作用や物体の材質などを考慮して動作する必要がある．

4. 本システムで扱う問題

本論文では，前章であげた問題のうち，特にユーザの視点が変化することによって生じる問題を扱う．本システムでは，ユーザはカメラロボットの映しだす映像を見ながら発話するが，ユーザはこのカメラ自体を移動させることもできる．そのためシステムはユーザの発話を，その視点を考慮して解釈する．また一般的な照応表現だけでなく直示的な表現を解釈することも可能である．

システムはまず指令を受けると，ユーザがどの視点にたってその指令を発話しているか，可能性のある視点を列挙する．次に，それぞれの視点からユーザがその指令を発したと仮定し，それぞれの解釈をスコア付けしたうえで最も妥当な解釈と思われるものを選ぶ．またユーザが「それ」などの代名詞を用いた場合，そ

れ以前の発話に適当な先行詞が見つからないときには，システムはユーザが直示を行っていると解釈する．先の処理によってユーザの視点が推定できるため，システムはそこから見た仮想世界の状況を考慮して直示的な表現を解釈することが可能となる．

本章では特に意味・照応解決モジュールに焦点をあて，格フレーム中のスロットに含まれた字句的な情報から，ユーザの視点を考慮し仮想世界上の実体を決定する手法を提案する．

4.1 本システムで使用する意味表現

最初に本システムで使用する意味表現を説明する．本システムでは，格フレーム中の名詞句もフレームによって表現されている．このフレームはさらに入れ子状になっており，その名詞句の構文的な構造を反映している．現在のところ，本システムでユーザが名詞句によって表現できる概念は，仮想世界上の位置あるいは物体のどちらかである．名詞句の構造は次のいずれかに限定されている：

- <形容詞> + <名詞>（例：赤い球，遠い場所）
- <位置を表す名詞句> + の + <物体を表す名詞>（例：右の球）
- <物体を表す名詞句> + の + <位置を表す名詞>（例：球の前）

構文解析モジュールは上のような規則によって名詞句の入れ子構造を生成する．フレーム生成モジュールは，この構文木をたどることによって入れ子状になったフレームを生成する．ここで，名詞句に対応するフレームは位置あるいは物体のどちらかを表現し，その入れ子構造は上に示したような名詞句の構造と一致したものになっている．さらに意味・照応解決モジュールがこれを再帰的にたどることにより，実際の仮想世界上の位置あるいは物体を決定する．このようにフレームを実際のオブジェクトへと変換する操作を，名詞句の解決と呼ぶ．以下はフレームによって名詞句がどのように表されるかを示した例である．

例：名詞句 “(遠くの (赤い 球))”：

```
Frame“ 遠くの赤い球 ”(物体)
  head:
    Frame“ 赤い球 ”(物体)
      head:
        Frame“ 球 ”(物体)
          class: 球
      modifier: 赤い
  modifier: 遠く
```

例：名詞句 “((その 机) の 右)”：

```
Frame“ その机の右 ”(位置)
  position-to:
    Frame“ その机 ”(物体)
      head:
        Frame“ 机 ”(物体)
          class: 机
      modifier: その
  head: 右 (位置)
```

入れ子状になったフレームの head スロットは名詞句のヘッドを表し，修飾語は modifier や position-to スロットによって表される．modifier スロットは物体を表すフレームに含まれ，head スロットのフレームによって表された名詞句を限定する働きを持つ．一方，position-to スロットは位置を表すフレームに含まれ，head スロットのフレームによって表された位置の表現が，何に対してのものなのかを示している．

4.2 ユーザの視点の決定

本システムでは，物体を表す名詞句には「その右にある球」「きみの前にある家」などのように，空間的な制約の表現がともなって現れることが多い．このような名詞句を解決するためには，まずその指令がどの視点から発されたのかを決定する必要がある．たとえばユーザがロボット A に対して「右の球を押せ」と指示した場合を考える．ユーザは仮想世界を表示した画面上のウインドウを見ながら発話する．そのためユーザの指令の視点は基本的にはカメラと同じである．しかしここに現れる位置表現「右」とは，ユーザから見て右であるのか，その指示を受けたロボット A から見て右であるのかが曖昧である．本システムでは，このような物体を表す名詞句にかかる位置表現を次の 2 通りの可能性で解釈する．

(1) その指令が，その指令を受けるロボットの視点から述べられている．
(2) その指令が，ユーザ（カメラ）の視点から述べられている．

本システムは，指定された物体が実際にその解釈の位置に存在しているかどうかを調べることによって，その表現の曖昧性を解消する．最初に (1) の解釈が試され，これにあてはまる物体が仮想世界上に存在しない場合は (2) の解釈が試される．しかし実際のところ，この順序は自明ではない．場合によってはどちらの可能性ともとれる表現もありうるからである．現在のところ，そのような状況では (1) の解釈が優先して使われる．実は上の例文にはもう 1 カ所，曖昧な部分がある．それは「何に対しての『右』なのか」と

いう位置表現の基点の曖昧さである．この場合，本システムでは基点となる名詞句が省略されているものとしてそれにふさわしい点を補う．省略の解決については 4.4 節で述べる．

一方,「その右へ行って」などの指令に現れる「その右」のような位置を表す名詞句を解釈する場合，その位置に物体が存在しているとは限らないため，上に示したような曖昧性解消手法は使えない．このような名詞句では，(1) の解釈における視点が優先して使われる．

4.3 特定の視点から発話された名詞句の解決

視点が特定できたら，システムは与えられた名詞句フレームをその視点から見たものとして解釈する．この方法は，対象となる名詞句の種類によって異なる．前章で述べたように，物体を表す名詞句の場合，システムは仮想世界上のすべての物体を探索し，その解釈にあてはまる物体が実際に仮想世界上に存在するかどうかによって解釈の妥当性を判断する．一方位置を表す名詞句の場合，システムはその名詞句の特徴から仮想世界上のある 1 点を直接算出する．以下，それぞれの種類ごとに説明する．

物体を表す名詞句の場合

まずシステムは与えられた名詞句から,「仮想世界上のあるオブジェクトが，その名詞句の表している物体であるかどうか」を判定する手続きを生成する．この手続きは仮想世界のオブジェクトを引数にとり，与えられた名詞句に対するそのオブジェクトの適合度を返り値とする λ 式の形になっている．本システムが受けつける「赤い」,「右にある」などの表現は，原始的な判定手続きを表す λ 式として辞書に格納されている．この λ 式はオブジェクトのほかに，それが解釈される際の視点も受け取るようになっており，その視点から見た適合度を返す．システムはこの λ 式を仮想世界のオブジェクトすべてに適用し，条件と合致するオブジェクトを選び出す．例として「右の球」という名詞句を解決することを考える．

名詞句 “右の 球”：

```
head:
    Frame“ 球 ”(物体)
      class: 球
modifier: 右の
```

辞書に含まれる λ 式：

球	:	$\lambda p.\lambda obj.$（オブジェクト *obj* が球ならば 1，そうでなければ 0）
右の	:	$\lambda p.\lambda obj.$（オブジェクト *obj* が視点 *p* から見て右にある度合い）

このように入れ子状になっているフレームの場合，システムは内側のフレームから探索していく．まず仮想世界のすべてのオブジェクトを対象として，このフレームの一番内側にある head スロットが表しているオブジェクトを探索する．この例では class スロットに「球」が含まれているので，辞書中の「球」に対応する λ 式が使われ，探索結果として仮想世界上のすべての球オブジェクトが得られる．次に，見つかったオブジェクトに対して modifier スロットの判定手続きを適用し，その候補をしぼりこむ．なお，判定手続きを呼び出す際にシステムは「それがどの視点から解釈されるべきか」という視点の位置もその λ 式に渡すようになっている．この例における適合度（右にある度合い）は，まずその視点から見た代表的な「右」という点を計算し，そこから当該オブジェクトまでの距離を求めることで算出している．このような計算手続きは λ 式の中に埋めこむことができるため，視点によって異なった解釈を単一の辞書項目で表現することが可能になる．このようにして得られた名詞句の対象が複数ある場合，システムはそれに付与された適合度の最も高いものを選び，最終的に一意のオブジェクトを得る．

位置を表す名詞句の場合

現在のシステムでは，位置は最終的に仮想世界上の 1 つの点と見なされる．しかし仮想世界上の点は無限にあるので，システムは物体を特定する場合のように仮想世界上の候補すべてを探索するわけにはいかない．そこでシステムはまず position-to スロットで表されているオブジェクトを再帰的に解決し，その位置を得る．次に，その位置に対して head スロットで表されている位置関係にある点を算出する．位置関係を表す語には，それを算出する手続きが対応づけられている．しかしこのような位置関係の解釈は，その表現を解釈する主体と対象となる物体の種類によって異なる．本システムの環境では，ユーザは基点となるオブジェクトの種類によって異なった視点を使うことがある．たとえば**図 3** のような状況では，向きのない球を基点とした位置関係はユーザの立場から解釈する必要があるのに対して（図 3-a），向きのあるロボットを基点とした位置関係はロボットの立場から解釈する必要がある（図 3-b）．さらに，球を基点とした左右の関係がロボットを基点としたものと逆になっている．本システムではこのような状況に対応するため，仮想世界上の点を算出する手続きに，視点の座標および位置関係の基点となるオブジェクトを引数として渡す．この手続きはヒューリスティックなルールを使うことでユー

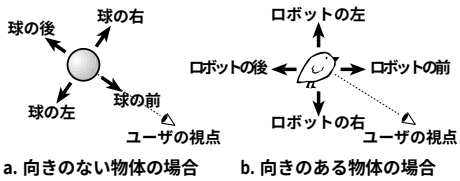

図 3 視点と対象によって異なる位置表現
Fig. 3 Spatial expressions.

ザにとって図 3 に示したような解釈の点を計算する．

4.4 照応・省略の解決

名詞句のなかには「それ」や「その球」などといった，代名詞や連体詞が含まれるものがある．このような語が名詞句中に現れると意味・照応解決モジュールはこれを照応表現であると見なし，照応解決のための手続きを実行する．

Grosz らによれば，ユーザが照応表現を用いるのは現在の文の焦点となる名詞句を表すためである[6),9)]．本システムでは，ユーザはあることをソフトウエアロボットに行わせるために，そのロボットに自分が望む仮想世界の状態，すなわち「ゴール」を伝えている，と見なすことができる．このような状況では，ユーザの焦点はそのユーザがこれから達成しようとしているゴールによっても変化する[7)]．一般的に，ユーザの望むゴールは 1 回の発話ですべて表現できるわけではない．そのためユーザは複数回の発話によって 1 つのゴールを表現するが，このようなときに照応表現が用いられることがある．そこで本システムではユーザのゴールを推測し保持することでユーザが用いる照応表現の参照先を決定する．

実際には，システムはゴールそのものではなく，同一のゴールを表現する一連の発話列を扱う．この発話列を発話スレッドと呼ぶ．本システムでは対話中のある瞬間に，複数の発話スレッドが同時に存在しうる状況を想定している（**図 4**）．本システムは発話スレッドを発話履歴データベース内に保持しており，ある発話がなされたときにそれが既存のスレッドを受けたものであるのか，あるいは新たなスレッドの生成を示すものであるのかを判定する．これは，その発話が既存のスレッドのどれかと一貫性を持っているかどうかによって判断する．一貫性の判定は，主語や動詞の一致，および手がかり句の存在などを考慮して行う．ユーザが照応表現を用いる場合，その指示対象はユーザが表現したがっているゴールに属する発話スレッド中にすでに現れているはずである．そのため，ユーザの発話に照応や省略が含まれていたり，その内容に前回の続きを示唆するような表現が含まれていたりする場合，

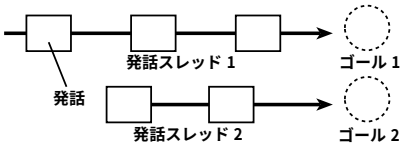

図 4 発話スレッド
Fig. 4 Utterance thread.

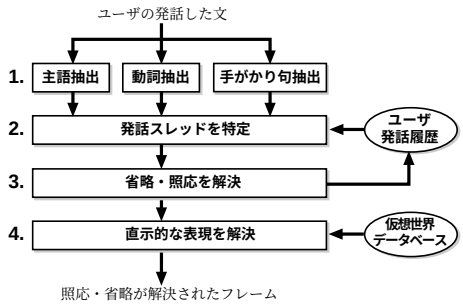

図 5 照応・省略解決のアルゴリズム
Fig. 5 Anaphora/ellipsis resolution.

システムは一貫性のあるスレッドを探索し，そのスレッドにあるこれまでの発話のフレームを使って照応および省略が解決できる．ユーザの発話に照応や省略が含まれていなかったり，あるいは一貫性のあるスレッドが見つからなかったりする場合，システムはそれを新規のスレッド生成と解釈する．

次に本システムにおける照応・省略解決の手順を示す（**図 5**）：

(1) ユーザの発話した文から，まず主語や動詞，および手がかり句を探索する．手がかり句とは「そのまま」「もうすこし」などの副詞句で，これはユーザが同一ゴールを指定するときの目印になることが多い．

(2) 次にユーザが発話した文とこれまでの発話スレッドの文とを比較し，ユーザのゴールを表現しているとみられるスレッドを探索する．この探索は最も新しい発話を持つスレッドから順に行われる．主語および動詞の両方が一致している文がスレッド中にあれば，そのスレッドが一貫性を持つものとして選ばれる．そのような文がなくとも「もっと」「そのまま」などの手がかり句が文中に存在し，なおかつ主語あるいは動詞が一致している文があれば，そのスレッドがユーザのゴールを表現していると見なされる．

(3) ユーザの発話と一貫性のある発話スレッドが特定されると，システムは同一スレッド上にある過去の文を取り出す．システムはこの文から照応表現の参照先を決定し，新しい文を追加して

発話スレッドを最新の状態に更新する．このようなスレッドが発話履歴データベース中に存在しない場合，システムはこの発話を新規のスレッド生成と見なし，新しいスレッドをデータベース中に作成する．

(4) 本システムでは，ユーザはこれまでの文に一度も表れていないものに対しても照応表現を使うことがある．たとえばユーザの目の前にあるものを「それ」などの代名詞によって直示することができる．一貫性のある発話スレッドが見つからない場合，システムはユーザが直示的な表現を使っていると解釈し，ユーザ（カメラ）の視界とソフトウエアロボットの視界を考慮して仮想世界上のオブジェクトを決定する．これは4.3節で述べた方法と同様に，仮想世界上のすべてのオブジェクトを探索し，ユーザの視点からみて最も近い場所にあるオブジェクトを直示の適合度が高いとして選び出すようになっている．

また，本システムでは文中の省略も解決することができる．ユーザは同一ゴールを修正する際には，前回の発話で指定したものについては省略するかもしれない．しかしひとたびユーザのゴールが特定できれば，省略されている名詞句に関しても照応表現と同じように前回の発話から自動的に補うことができる．本システムでは現在これを格スロット単位で行っており，次のような発話に現れる省略を解決できる：

(1) 「それを押して」

(2) 「もっと [それを] 押して」

この例では [～] 内が省略されており，システムは不足している格を補完することによって指令を実行する．4.2節で述べたようなある位置表現の基点となる名詞句が省略されている場合，システムは現在の焦点の目的格を省略された基点として計算する．このような焦点が存在しない場合には，ユーザの視点が基点として計算される．この規則によって，たとえば単に「右から映せ」という指令が来た場合は，その基点をユーザと解釈し,「その球を右から映せ」という指令が来た場合は，その基点を球と解釈する，すなわち「その球をその球の右から映せ」と解釈することが可能になる．

この手法では，省略された名詞句はすでに仮想空間中に存在しているか，あるいは対話中に最低一度は現れていなければならない．しかしたとえば「後に下がれ」などの文では,「後」という名詞句は自明であり省略できる．このような省略は現在のところアドホックなルールによって処理されている．

5. おわりに

本論文では自然言語を用いて仮想世界を操作するインタフェースの利点および問題点をあげ，これらを研究するためのプラットフォームとして我々が開発しているシステム傀儡について述べた．次にユーザの仮想世界上の位置や物体を表現する名詞句を解決し，仮想世界上のオブジェクトを一意に決定するための手法を提案した．ユーザはある動作を指令するときに，自分の持っているゴールを伝達しようと試みる．本システムは発話スレッドを用いることでユーザのゴールを推測し，内部のデータベースを更新する．これによってユーザが現在どの物体あるいは位置に焦点を置いているかが推定できる．

ユーザはカメラを通して仮想世界の映像を見ながら発話するため，その表現の解釈はユーザの視点によって変わる．また，ユーザは直示を用いる場合もある．このような場合，システムはユーザの視界および仮想世界の状況を考慮し，その物体を特定する．これによって照応表現や省略が含まれる文に対してもユーザの意図した動作を適切に実行することができる．

本システムが正しく扱える表現はまだ限られている．4.4節で述べた発話スレッドによる焦点の推測は，文の比較的表層的な面しか考慮に入れていない．たとえばある発話が同一のスレッドに属するかどうかを判定するのに，それらの動詞が同じであるかなどの情報をもとに判断している．しかし，ときにユーザは暗黙の焦点の移動を行う場合がある．たとえばある動作を行ったあとは，次にくる動作が予想できる場合などである．このような場合，ユーザはその動詞を想定して照応あるいは省略を行うため，本システムでは発話スレッドの識別に失敗し，照応や省略を正しく解決できない．ユーザの視覚による焦点の移動もある．本システムでは指令の動作主が省略された場合，基本的にはその1つ前の指令を実行したロボットがその指令を受けると解釈する．だがユーザがカメラの向きを変えると，カメラの前には新しい物体が現れ，ユーザの焦点はその新しく現れた物体に移動することが多い．現在のところ，このような焦点の移動にも追随できない．

また，本システムでは係り受けの曖昧性解消を行っていない．そのため，たとえば「右にある球の左にある球」などの表現では「右にあり，同時にかつ何かの球の左にある球」として解釈すべきなのか「右にある何らかの球に対して左にある球」として解釈すべきなのかを判断できない．このように本質的に曖昧な指令を受けた場合，システムはユーザに問い返すべきであ

るが，現在のシステムではどちらか一方の解釈に決められてしまう．またユーザが発話する文は命令文だけであると仮定し，疑問文や叙述文を扱えない．これらの問題に加えて，3 章であげたさまざまな問題を解決していくことが今後のおもな課題である．今後，仮想世界の構造を複雑化し，ユーザの指示できる範囲を広げる予定である．このような拡張をより自然に行えるようなアーキテクチャを提案することも重要な課題となる．

参 考 文 献

1) Allen, J.F. and Perrault, C.R.: Analyzing Intention in Utterances, Grosz, B.J., Jones, K.S. and Webber, B.L.(Eds.), *Readings in Natural Language Processing*, pp.441–458, Morgan Kaufmann Publishers Inc., ISBN 0-934613-11-7 (1986).
2) Austin, J.L.（著），坂本百大（訳）：言語と行為，大修館書店 (1978).
3) Cohen, P.R. and Perrault, C.R.: Elements of a Plan-Based Theory of Speech Acts, Grosz, B.J., Jones, K.S. and Webber, B.L.(Eds.), *Readings in Natural Language Processing*, pp.423–440, Morgan Kaufmann Publishers Inc., ISBN 0-934613-11-7 (1986).
4) Fillmore, C.J.（著），田中春美，船城道雄（訳）：格文法の原理，三省堂，ISBN 4-385-30085-2 (1975).
5) Geib, C.W., Levison, L. and Moore, M.B.: SodaJack: An architecture for agents that search for and manipulate objects, Technical Report MS-CIS-94-16/LINC LAB 265 (1994).
6) Grosz, B.J., Joshi, A.K. and Weinstrin, S.: Providing a Univied Account of Definite Noun Phrases in Discourse, *Proc. ACL*, pp.44–49 (1983).
7) Grosz, B.J. and Sidner, C.L.: Attention, Intentions, and the Structure of Discourse, *Computational Linguistics*, Vol.12, No.3, pp.175–204 (1986).
8) Herskovits, A.（著），堂下修司，西田豊明，山田 篤（共訳）：空間認知と言語理解，オーム社，(1991).
9) Sidner, C.L.: Focusing in the Comprehension of Definite Anaphora, Brady, M. and Berwick, R.C.(Eds.), *Computational Models of Discourse*, MIT Press (1983).
10) Strassmann, S.: Semi-Autonomous Animated Actors, *Proc. 12th National Conference on Artificial Intelligence*, pp.128–134 (1994).
11) Winograd, T.: *Understanding Natural Language*, Academic Press (1972).
12) 片桐恭弘：談話の世界，自然言語理解，田中穂積，辻井潤一（編），pp.159–190, オーム社，ISBN4-274-07398-X (1988).
13) 国立国語研究所：日本語の指示詞，国立国語研究所 (1981).
14) 佐藤泰介，田中穂積，渕 一博：VISUALIZER — 自然言語理解システムの立場からみた機械による空間の把握，電子通信学会誌 (1976).

(平成 12 年 10 月 31 日受付)
(平成 13 年 4 月 6 日採録)

新山 祐介

2000 年東京工業大学情報理工学研究科計算工学専攻修士課程修了．現在，東京工業大学情報理工学研究科計算工学専攻技術補佐員．人工知能学会会員．

徳永 健伸（正会員）1983 年東京工業大学工学部情報工学科卒業．1985 年同大学院理工学研究科修士課程修了．同年（株）三菱総合研究所入社．1986 年東京工業大学大学院博士課程入学．現在，同大学院情報理工学研究科助教授．自然言語処理，計算言語学の研究に従事．工学博士．認知科学会，人工知能学会，言語処理学会，計量国語学会，Association for Computational Linguistics 各会員．

田中 穂積（正会員）

1964 年東京工業大学理工学部制御工学科卒業．1966 年同大学院修士課程修了．同年電気試験所（現，電子技術総合研究所）入所．1983 年より東京工業大学工学部助教授．現在，同大学院情報理工学研究科教授．自然言語処理，人工知能に関する研究に従事．工学博士．電子情報通信学会，認知科学会，人工知能学会，計量国語学会，言語処理学会，Association for Computational Linguistics 各会員．