エージェント対話システムのための対話処理と応答文生成

Dialog Processing and Output Generation for a Agent Dialog System

多胡順司† 広瀬啓吉‡ 峯松信明§

Junji Tago Keikichi Hirose Nobuaki Minematsu

1. はじめに

音声認識･音声合成技術の進歩に伴い，音声対話システムの研究が盛んに行われるようになった．音声対話システムはユーザの発話を入力とし，そこからユーザの意図を汲み取り，それに従った処理を行ったり，ユーザに対して文音声として情報を伝えたりするものである．

多くの対話システムは音声出力に、既存のテキスト音声合成（TTS：Text To Speech）器を用いている．これはテキストから朗読音声を合成することを目的として作られたものであり，より自然な対話音声を合成するためには、談話情報などの高次の言語情報に基づいて韻律を制御できる音声合成器が必要である．そこで概念音声合成（CTS ： Concept To Speech）により応答文の生成を行うことが提案されている[1]．TTS がテキストを入力とするのに対し，CTS ではシステムの内部表現（概念）から直接音声を合成する．CTS では文の生成過程で正確な言語情報が得られるため、統語構造を韻律に反映させたり、談話情報で韻律の焦点を制御したりすることが容易に行える。

また，多くの対話システムは静的な情報を扱うのみにとどまっている．一方で，動的な空間においてソフトウェアロボットの行動を自然言語で制御する研究が行われている[2]が，これらの研究はソフトウェアに指示を与えるだけでソフトウェアロボットとの対話は行われない．

そこで筆者らは，エージェントを用いて仮想空間中の物体を操作するエージェント対話システムを構築した．ユーザはエージェントに自然言語で指示を行うのだが，その過程でエージェント自身では問題が解決できない場合はユーザとの対話を通して問題を解決する．

2. 対話システムの構成

対話システムの構成を図 1 に示す．**音声認識部**はユーザの発話を入力としてそれを文字列に変換して出力する．**構文解析部**は文字列の形態素解析・構文解析を行い構文木構造として出力する．**対話管理部**は構文木構造として受け取ったユーザの発話からユーザの意図を判断し，空間の状態を考慮しながらエージェントの動作命令を送ったり，ユーザと対話するために韻律制御記号を含む音素記号列を音声合成部に出力したりする．**音声合成部**は韻律制御記号を含む音素記号列から，基本周波数パターン生成過程モデルに基づいて音声合成を行う[1]．**仮想空間管理部**は空間の状態を管理し，対話管理部のリクエストに応じてエージェントを動かしたり空間の状態を対話管理部に返したりする．**CG 生成部**は空間の状態をリアルタイムに表示する．

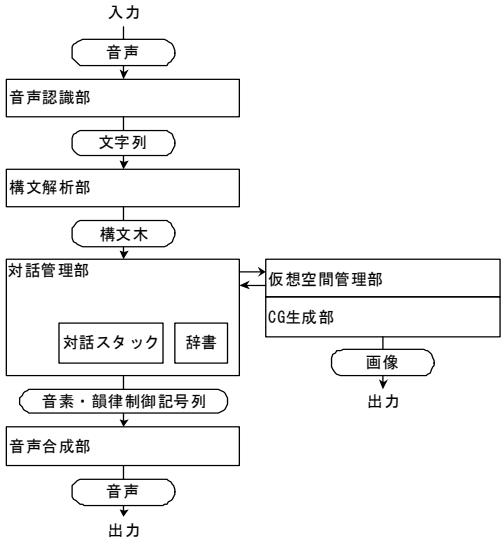

図 1 対話システムの構成

3. 対話管理部での言語情報の取扱い

3.1. 辞書

辞書はユーザの発話文の理解と応答文生成に用いる．本対話システムは品詞辞書・活用辞書・単語辞書の 3 種類の辞書を持つ．辞書は拡張性を考慮して XML で記述した．

3.1.1. 品詞辞書

品詞辞書は個々の品詞について以下の情報を定義する．

name 品詞名
independence 自立語・付属語
connection 接続

品詞は入れ子状にすることができ，省略した情報は親の品詞のそれを継承する．

†東京大学大学院工学系研究科
‡東京大学大学院新領域創成科学研究科
§東京大学大学院情報理工学系研究科

3.1.2. 活用辞書

活用辞書は個々の活用型について以下の情報を定義する．

name 活用型名
form 活用形（複数持つことができる）

さらに活用形（form）は以下の情報を持つ．

name 活用形名
display 活用形の表示用文字列
phoneme 活用形の発音する場合の音素記号列

3.1.3. 単語辞書

単語辞書は個々の単語を定義する．単語辞書は以下のように表される．（例：「移動する」）

```
<node>
  <identifier>移動する</identifier>
  <part>動詞<part>自立</part></part>
  <stem>移動</stem>
  <phoneme>i F@ do u</phoneme>
  <inflection>サ変・ースル</inflection>
  <dialog_data>
    <identity>move</identity>
    <attribute>agent_action</attribute>
  </dialog_data>
</node>
```

単語辞書は個々の単語について以下の情報を定義する．

identifier 単語を特定するための文字列
display 単語を表示する場合の文字列
part 単語の品詞
stem 単語の語幹（活用語のみ）
inflection 単語の活用型（活用語のみ）
connection 単語の接続
phoneme 単語の音素記号（アクセント指令を含む）
dialog_data 対話システム固有のデータ（対話用データ）

3.2. 単語･文章の表現

対話管理部では言語情報を一貫して構文木構造で扱う．

3.2.1. 言語情報の入力

対話管理部への文の入力に際しては構文木情報を LISP 形式で表現する．例えば「イスを机の前に置いて」という文の構文木構造は図 2 に示すとおりである．このとき LISP 形式では「(て (置く (を (イス))(に (前 (の (机))))))」と表すことができる．

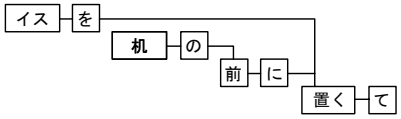

図 2 「イスを机の前に置いて」の構文木構造

また，単語にタグを与えたり単語の代わりにタグを用いたりしておくことで，文の単語にアクセスしたり文を接続したりすることができる．例えば「**アイテム**を**場所**に置く」という文は LISP 形式で「(置く$PRED (を ($ITEM)) (に ($POS)))」のように$PRED，$ITEM，$POS というタグを埋め込んでおくことで，$ITEM，$POS タグの部分に単語や，句を接続したり，$PRED タグを参照することで述語にアクセスしたりすることができる．

3.2.2. 文の作成

文の作成は単語を構文木構造にしたがって接続することで実現する．日本語における活用は，その語の掛かっている単語の接続によって決まる．すなわち，構文木構造に従って単語を接続していくことで自動的に活用形まで決定することができる．例えば「(て (持つ))」という構文木構造の場合，活用語である「持つ」は「て」に掛かっている．「て」の接続は「連用タ接続」で，「持つ」の活用型は「五段・タ行」で語幹は「持」，「五段・タ行」活用型の「連用タ接続」は「っ」である．したがって，「(て (持つ))」は「持って」と表示することができる．

さらに，タグを参照することで単語の重要度を設定し，重要度・構文木構造から韻律制御パラメータを設定し音声を合成する．

3.3. 音声合成

本対話システムで用いる音声合成器は文献[1]の規則に従う．この対話音声のための韻律規則では構文情報が与えられていれば，アクセント指令の位置と，単語の新しさ・重要度を設定するだけで韻律の制御が可能となる．本対話システムでは基本的には自立語にはアクセント指令を設定し，それを辞書に記述する．単語の新しさ・重要度はタグを参照することで合成のつど文ごとに設定する．

4. 対話処理

4.1. 対話システムの扱う項目

4.1.1. アイテム

アイテムとは，仮想空間中に置かれたもののことである．アイテムは図 3 のようなものがある．アイテムは種類・色の属性を持つ．図 3 の左のアイテムの場合は種類が「電話」，色が「赤」で右が「イス」，「灰色」である．

図 3 アイテムの例

4.1.2. 場所

場所は仮想空間中の位置を表すものである．空間はグリッドに区切られており，場所はそのグリッド座標で扱われる．

4.2. 対話処理

4.2.1. 対話管理部の処理

対話管理部での処理の流れは図 4 に示すとおりである．初期状態から構文木情報を含んだ発話文を受け取ると，その内容に応じて処理を切り替える．現在，処理できるのはエージェントへの命令だけであるが，新しい機能を容易に付け加えることができる．

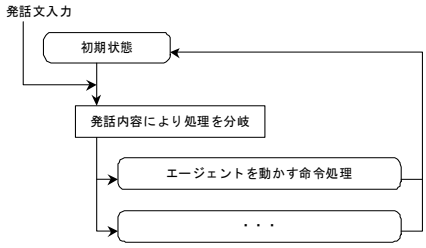

図 4 対話管理部での処理の流れ

4.2.2. エージェントへの命令の処理

エージェントへの命令の処理を図 5 に示す．まず，発話文から命令の種類を判別する．この命令をもとに格解析により発話文中でアイテム・場所を表す句を探し，それの指すものを特定する．続いて，エージェントの動作を決定する．「アイテム・場所の特定」，「エージェントの動作の決定」のそれぞれの段階で，システム自身で問題を解決できない場合はユーザとの対話を通して問題を解決する．

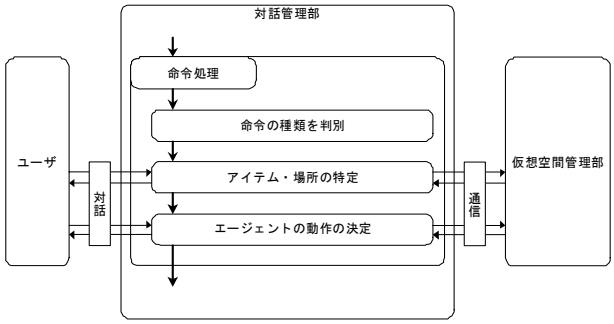

図 5 エージェントへの命令の処理の流れ

4.2.3. アイテム・場所の特定

ユーザが「机」といっても，仮想空間中に机が複数ある場合はユーザがどの机を指しているのかを特定しなくてはならない．基本的にはユーザの発話文の構文木の枝側から決定していく．たとえば「冷蔵庫の前の机を持って」という入力文の場合，構文木は図 6 のようになる．このとき，「冷蔵庫」というアイテムを特定し，続いて「冷蔵庫の前」という場所を特定，さらに「冷蔵庫の前の机」というアイテムを特定する．特定の際に，候補が複数あったり，該当するアイテム・場所が見つからなかったりする場合はユーザとの対話を通して問題を解決する．

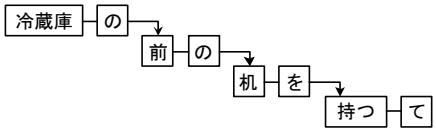

図 6 「冷蔵庫の前の机を持って」の構文木構造

4.2.4. エージェントの動作の決定

「状態」・「動作」・「命令」を定義し，それぞれに自身を表す言語情報を与えておく．「状態」は空間の状態を表すものであり，表 1 のようなものがある．「動作」は状態を前提状態から終了状態に変化させるものであり，表 2 のようなものがある．「命令」はエージェントに状態を目標状態に変化させるものであり,表 3 のようなものがある．

表 1 「状態」の例

状態
アイテムの前に移動できる
アイテムが目の前にある
アイテムを持っている
アイテムが場所にある

表 2「動作」の例

動作	前提状態	終了状態
アイテムの前に移動する	アイテムの前に移動できる	アイテムが目の前にある
アイテムを持つ	アイテムが目の前にある	アイテムを持っている

表 3 「命令」の例

命令	目標状態
持つ	アイテムを持っている
置く	アイテムが場所にある

エージェントの動作を決定するには，まずユーザの発話から「命令」を抽出する．そして「命令」から目標状態を設定する．続いて，設定した目標状態を終了状態とする「動作」を検索し，その「動作」の前提状態が満たされているかを判断する．満たされていれば動作を実行する．満たされていなければその前提状態を新しい目標状態とする．以上を再帰的に繰り返し，すべての目標状態を実現することでユーザの命令を実行する．このとき，エージェント自身では解決できない場合はユーザと対話を行うことで問題を解決する．

4.3. ユーザの発話文の理解

4.3.1. 対話用データ

対話用データは単語に関連付けられている対話システム依存の情報である．本対話システムでは対話用データには表 4 に示すようなものがある．

表 4 対話用データの例

attribute	identity	内容
item_type		アイテムの種類を表す単語
	desk	机
	shelf	棚
item_color		アイテムの色を表す単語
	red	赤
	blue	青
position		場所を表す単語
	front	前
	left	左
agent_action		エージェントの動作を表す単語
	take	持つ

4.3.2. ユーザの発話の理解

ユーザの発話文の対話用データを参照することでユーザの発話を理解する．発話文のルートとなる文節の自立語の対話用データがエージェントの動作を表す単語であればエージェントへの命令と判断する．

4.4. 「アイテム・場所の特定」での応答文生成

検索結果を言語情報に変換して，文を接続する．アイテムの特定での文の接続は図 7 のように行う．ユーザの発話文のうち，アイテムを表す部分（ここでは「電話」）を構文木構造のまま取り出し，検索結果を言語情報に変換した文の$ITEM タグに接続する．つづいて文を接続するための「$SNTC のですが」という文に接続し，最後にユーザの応答を促すための文「$SNTC どれのことですか」という文に接続して，「電話はいくつかあるのですがどれのことですか」という応答文を生成する．場所の場合も同様である．

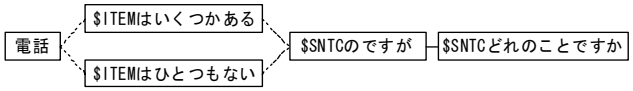

図 7 アイテムの特定での応答文生成法

4.5. 「エージェントの動作の決定」での応答文生成

前述したとおり，「状態」・「動作」・「命令」はそれぞれ自身を表す言語情報を持っている．例えば，「置く」という「動作」は，「$ITEM を$POS に置く」という言語情報を持っている．したがって，エージェントの動作の決定での応答文生成はエージェントの思考過程すべてを文にすることも可能である．その例を図 8 に示す．

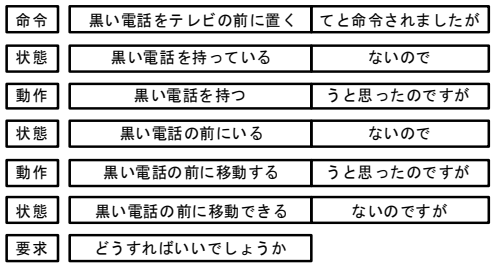

図 8 システムの思考過程の表示

この文は非常に冗長なので，実際のシステムでは解決できない状態と要求だけを出力する．すなわち，「黒い電話の前に移動できないのですが，どうすればいいでしょうか」となる．

5. エージェント音声対話システム

対話システムの画像出力を図 9 に示す．また対話例を以下に示す．U はユーザ，S はシステムである．

U1:電話を置いて
S1:電話はいくつかあるのですがどれのことですか
U2:黒い電話です
S2:どこに置けばいいのですか
U2:テレビの前に置いて
S2:電話の所に移動できないのですがどうすればいいでしょうか
U3:花瓶をパソコンの前に置いて

図 9 画像出力

6. まとめ

対話管理部での言語情報を一貫して構文木構造で扱うことで音声合成との親和性の高い応答生成を実現した．内部状態（概念）に自身を表す言語情報を与えておき，応答文生成の際にそれらを接続することで柔軟な応答文生成を実現した．

今後は辞書の充実，韻律のさらなる細かな制御などが課題である．また，内部表現をより詳細に設定することでさらに柔軟な応答文生成を実現する．

参考文献

[1]. 桐山伸也, 他``応答生成に着目した学術文献検索音声対話システムの構築とその評価", 信学論 D-II, Vol.J83-D-II, No.11, pp2318-2329(2000)
[2]. Y. Shinyama, et. al, ``Kairai – Software Robots Understanding Natural Language", Third International Workshop on Human-Computer Conversation (2000)
[3]. Keikichi Hirose, et al, ``Synthesizing dialogue speech of Japanese based on the quantitative of prosodic features, " Proc. ICSLP96, vol.1, pp.378-381(1996)