社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－SLP－47 (13)
2003／7／19

エージェント対話システムにおける対話管理と応答生成

八木 裕司†, 多胡 順司†, 峯松 信明††, 広瀬 啓吉‡

† 東京大学大学院 工学系研究科, †† 東京大学大学院 情報理工学系研究科,
‡ 東京大学大学院 新領域創成科学研究科
Tel.: 03-5841-6393, Fax.: 03-5841-6648
{yagi, tago, mine, hirose}@gavo.t.u-tokyo.ac.jp

あらまし 音声認識や自然言語処理、音声合成技術の進歩に伴い、数多くの音声対話システムが構築されてきている。しかしながら、音声対話システムの多くは、音声合成部に既存のテキスト音声合成器を用いているため、応答生成の過程で得られる言語情報を音声出力に反映させることができない。そこで我々は、概念音声合成（CTS : Concept-To-Speech）をエージェント対話システムで実現した。このシステムは、ユーザとの対話を行ないながら、エージェントを用いて仮想空間中の物体を操作するものである。本システムではCTSの実現のために、対話処理において言語情報は一貫して構文木構造で扱われる。さらに、対話管理手法もCTS用に作られている。

キーワード 音声対話システム、対話管理、応答生成、概念音声合成、言語情報

Dialog Management and Response Generation for an Agent Dialog System

Yuji Yagi†, Junji Tago†, Nobuaki Minematsu†† and Keikichi Hirose‡

† Graduate School of Engineering, University of Tokyo,
†† Graduate School of Information Science and Technology, University of Tokyo,
‡ Graduate School of Frontier Sciences, University of Tokyo
Tel.: 03-5841-6393, Fax.: 03-5841-6648
{yagi, tago, mine, hirose}@gavo.t.u-tokyo.ac.jp

Abstract Due to recent developments in natural language processing, speech recognition and speech synthesis technologies, a rather large number of spoken dialog systems have been constructed. However, most of those systems simply used text-to-speech conversion modules developed separately, and, therefore, failed to include rich linguistic information obtainable during text generation process into speech outputs. We have realized a concept-to-speech (CTS) conversion scheme in an agent dialog system, where an agent (a stuffed animal) walked around in a small room constructed on a computer display to complete some jobs with instructions from a user. In order to realize the CTS conversion, the ilnguistic information was handled as a tree structure in the whole dialog process. Also a dialog management scheme was developed suited to the CTS conversion.

Key words Spoken dialog system, Dialog management, Response Generation, CTS conversion, Linguistic information

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な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なったり、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では、音声出力に既存のテキスト音声合成器（TTS：Text-To-Speech）を用いている。しかし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、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な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）を利用して音声を合成する。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なえる。

また、多くの音声対話システムでは、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なわれている [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なわれない。そこで我々は、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なうのだが、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・構文解析を行ない、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への動作命令

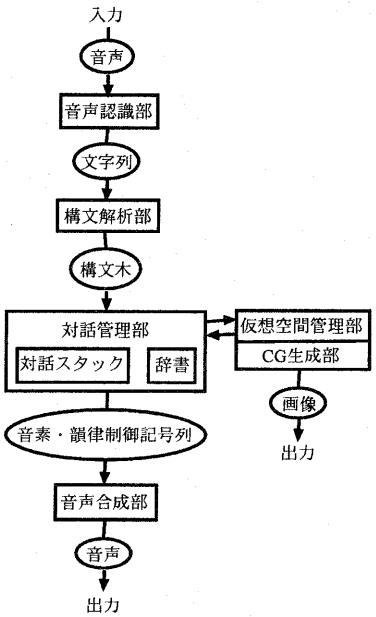

図 1. 対話システムの構成

を生成したり、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なう [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。

3 対話管理部での言語情報の取扱い

3.1 辞書

言語情報を取り扱う上で、辞書は必要不可欠なものである。そのため、構文解析部・対話管理部では、ユーザの発話文の理解と応答文生成に辞書を用いる。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。それぞれの辞書は、拡張性を考慮して XML で記述されている。

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。

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_symbol 単語の音素記号

dialog_data 対話システム固有のデータ

付属語には、これらの他に [3] によって得られたアクセント結合規則を付与した。なお、この規則は、付属語アクセント規則の他に、複合名詞や接頭辞に関する規則、文節間規則がある。

3.2 単語・文章の表現

CTS を実現するために、対話システムで扱う内部情報について言語情報を定義する。本対話システムでは対話管理部において、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。

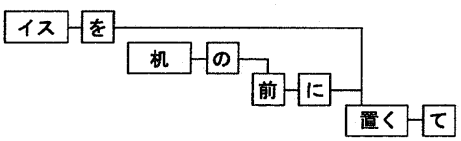

図 2. 「イスを机の前に置いて」の構文木構造

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 形式で表すと「（置く**$PRED**（を（**$ITEM**））（に（**$POS**））」となり、このように**$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従って接続することで実現する。日本語における活用は、その語のか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」にかかっている。「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。従って、「（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 [4] 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オブジェクトのことである。アイテムには図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。

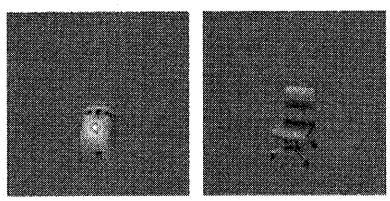

図 3. アイテムの例

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 エージェントへの命令の処理

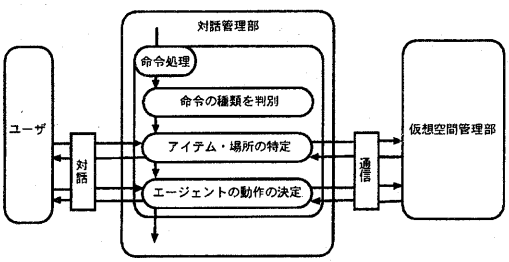

図 4. エージェントへの命令の処理の流れ

エージェントへの命令の処理を図4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。

4.2.2 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には、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。例えば「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5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。

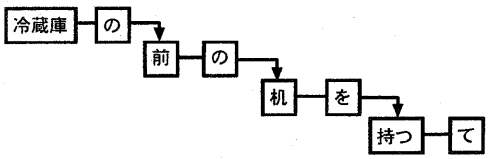

図 5. 「冷蔵庫の前の机を持って」の構文木構造

4.2.3 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。

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、全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問題を解決できない場合は、ユーザと対話を行なうことで問題を解決する。「状態」・「動作」・「命令」とも、辞書と同様に今後の拡張性を踏まえて、XML で記述されている。

表 1. 「状態」の例

名前	状態の表示（内容）
movablef_o	**アイテム**の前に移動できる
frontof_o	**アイテム**が目の前にある状態
have_o	**アイテム**を持っている状態
o_on_p	**アイテム**が**場所**にある状態

表 2. 「動作」の例

名前	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 手が空いている	アイテムを持っている

表 3. 「命令」の例

名前	目標状態
持つ（take）	**アイテム**を持っている
移動する（move）	**場所**が目の前にある
置く（put）	**アイテム**が**場所**にある

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。対話用データには表 4 に示すようなものがあり、ユーザ発話の理解と応答文生成に用いられる。

表 4. 対話用データの例

attribute	identity	内容
item_type		アイテムの種類
	desk	机
	shelf	棚
item_color		アイテムの色
	red	赤色
	blue	青色
position		場所
	up	上
	down	下
item		アイテム
	mono	もの
agent_action		エージェントの動作
	take	持つ
	put	置く

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 6 のように行なう。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**$ITEM** タグに接続する。続いて文を接続するための「**$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。

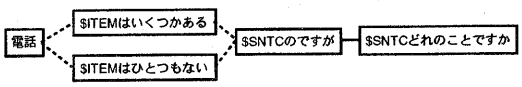

図 6. アイテムの特定での応答文生成法

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。従って、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全てを文にすることも可能である。その例を図 7 に示す。

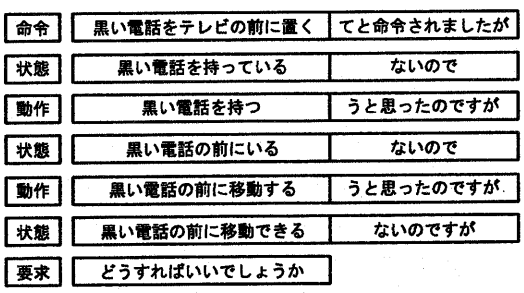

図 7. システムの思考過程の表示

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 エージェント対話システム [5, 6]

対話システムの画像出力を図 8 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: 電話をパソコンの前に置いて

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: 黒い電話の所に行けないのですが、どうすればよいでしょうか

U3: 花瓶を持って

S3: 手が空いていないのですが、どうすればよいでしょうか

U4: テレビの前に置いて

S4: 花瓶を置いていいですか

U5: はい

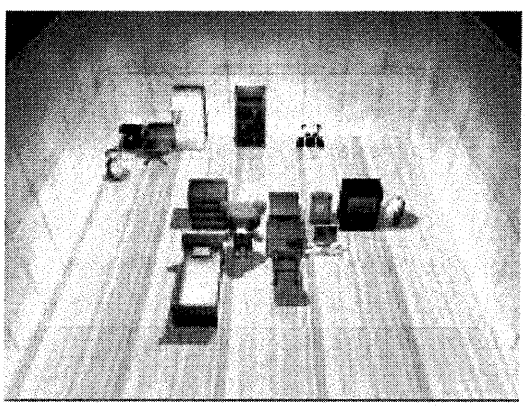

図 8. Output of the system

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。

今後の課題として、辞書の充実や韻律のさらなる細かな制御、省略・照応の解決などがある。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, 広瀬 啓吉: “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,” 電子情報通信学会論文誌 D-II, vol.J83-D-II, no.11, pp.2318-2319, 2000.

[2] Y. Shinyama, T. Tokunaga and H. Tanaka: “Kairai - Software Robots Understanding Natural Language,” Third International Workshop on Human-Computer Conversation, 2000.

[3] N. Minematsu, R. Kita and K. Hirose: “Automatic estimation of accentual attribute values of words to realize accent sandhi in Japanese text-to-speech conversion,” Proc. IEEE 2002 Workshop on Speech Synthesis, Santa Monica, 2002.

[4] K. Hirose, M. Sakata and H. Kawanami: “Synthesizing dialogue speech of Japanese based on the quantitive analysis of prosodic features,” IEICE trans. Fundamentals, Vol.E76-A, No.11, pp.1971-1980, 1993.

[5] 多胡 順司, 広瀬 啓吉, 峯松信明: “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,” 情報処理学会第 65 回全国大会, 5T7B-4, Vol.5, pp.507-510, 2003.

[6] K. Hirose, J. Tago and N. Minematsu: “Speech Generation from Concept for Realizing Conversation with an Agent in a Virtual Room,” Proceedings 8th European Conference on Speech Communication and Technology, Genova, to be published(2003-9).