人　工　知　能　82－3
（１９９２．　５．　２６）

音声言語理解のための創発的計算モデル

奥乃 博　　岡田美智男
日本電信電話 (株)　　NTT 基礎研究所

本稿では，音声言語研究に対する新しい計算モデルの提案を行う．　音声言語研究は，信号処理である音声認識と記号処理である自然言語処理とを結合することによって，従来よりも柔軟で知能的な処理能力を実現する新しい研究分野である．　本稿でのアイデアは，さまざまな能力を備えたエージェントの社会として音声言語システムを構築することである．　エージェントの社会とは，包摂アーキテクチャ (*subsumption architecture*) による inhibitor/suppressor ネットワークで結合されたエージェントの集合であり，inhibitor/suppressor の創発的計算 (*emergent computation*) の結果として，さまざまな音声言語活動が実現される．たとえば，「聞き流すこと」ができたり，「聞き耳を立てること」ができるような効果がシステムに与えられることになる．

Emergent Computation Model for Spoken Language Understanding

Hiroshi G. Okuno　　Michio Okada
NTT Basic Research Laboratories
3-9-11, Midori-cho, Musashino, Tokyo 180 JAPAN
okuno@pooh.ntt.jp　　*okada@atom.ntt.jp*

We propose a new computation model for spontaneous speech understanding. Since spontaneous speech includes various ill-formed sentences and human communication usually does not require complete recogniton of speech, adaptability and situated-orientedness is considered important for such system. The proposed architecture consists of a set of competent agents that have their own goals. Competent agents can activate or inhibit other agents in the sense of subsumption architecture proposed by Brooks and agents are fired by spreading activation in the sense of neural network proposed by Maes. The action taken by the system is emergent computation determined by situation, goals, activation/inhibition parameter.

1 はじめに

信号処理である音声認識と記号処理である自然言語処理との融合によって，より柔軟で知能的な処理能力を実現しようという音声言語の研究が最近活発に行われるようになってきた．とくに，**自然な発話** (*spontaneous speech*) を研究対象としていることが目立った特徴である．音声処理，自然言語処理の研究は，NFS が推進する High Performance Computing Project に掲げる Grand Chanllenges の一つとして取り上げられており，その課題として音声認識と自然言語の融合，コネクショニスト自然言語処理等への展開が主張されている [8].

従来の自然言語処理あるいは音声処理研究で研究対象となった発話は，限定された文法あるいは制限された語彙しか使えないコントロールされたものであった．さらに話者が特定されている場合すらあった．それに対して，自然な発話では，文法的に間違った文や言い直し，繰り返し，言い詰り，尻切とんぼ，などが含まれるので，その処理の複雑さはコントロールされた発話と比較して，はるかに大きい．自然な発話を取り扱うには，既存の自然言語処理と音声処理とを融合させるだけでは不十分であり，新しいアーキテクチャが必要である．

音声認識システムの従来の研究の流れは 2 つに大別できる [15]．一つは扱うコーパスの大規模化の研究である．DARPA の資金援助によって，膨大な語彙を使用した不特定話者の連続音声を高速，かつ高認識率で認識する研究が推進されてきた．共通のデータベースを使用し，認識率を競うという研究は評価基準が明確になっているので，素人目にも極めて分りやすい研究である．もう一つは，話し手の意図や発話プランを理解することが自然言語理解，あるいは音声認識で重要であるという人工知能研究の立場から音声言語を研究しようというアプローチである．

この 2 つの流れは音声言語研究への期待にどう答えるかの違いであるが，いずれも重要な研究の展開である．しかし，どちらか一方に組するのは不十分な結果しか得られないように思える．というのは，人間の音声によるコミュニケーション活動においては，必ずしも音声が 100% 認識されているとは思われない．適当に聞き流して他の事をしたり，あるいは聞き流しているかと思えば，突如話に耳を傾けるというような場面が少なからずある．常時 100% の認識率を誇る必要などはないのである．また，コミュニケーションにおいて，いつもその場で相手の発話意図・プランまで理解しているとはとても思えない．常時深い理解をするというよりは，その場その場の状況に応じて，音声認識，自然言語理解を行えばよいのである．

音声言語システムでは，以下のような機能が要求される．

(1) ゴール指向の処理 — 音声言語システムを構成する個々のエージェントは受動的な処理だけではなくゴールを陽に持つ，

(2) 状況指向 — 現在の状況に応じて，反射反応的な応答から慎重な応答と応答時間や処理レベルが変わる，

(3) 頑健さ — 誤りを含んだ入力，曖昧で不完全な入力，エージェントの一部が壊れてもシステム全体としては何らかの反応をする，

(4) 先読み — 音声言語システムを構成するエージェントの一部は，システム資源が空いているときには先読み等の処理を行い，より高度な知能処理を実現する．

本稿では，このような要求条件を満足するための新しいシステムアーキテクチャとして，さまざまな能力を備えたエージェントの社会 (集合) として音声言語システムを構築する方法を検討する．なお，提案するアーキテクチャの前提として我々が仮定するのは次の命題である．

> 適合的あるいは知的な性質の源は，マルチエージェントシステムでの競合するゴールの中から，状況や入力に応じて行動が定まるという創発的な計算にある．

以下，第 2 章で従来のシステムアーキテクチャを概観し，第 3 章でマルチエージェントシステムの課題を述べる．第 4 章で行動ネットワークについて，第 5 章で考察を行う．

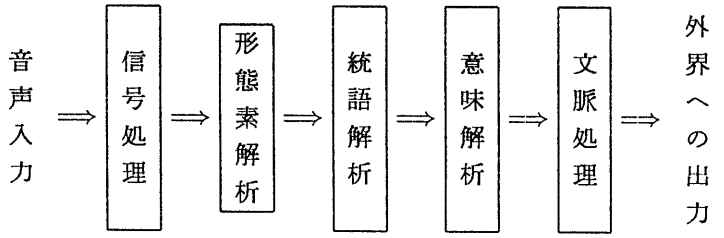

図 1: 従来の機能分割構成法による自然言語理解

2 音声言語システムアーキテクチャ概観

音声言語のためのシステムアーキテクチャとしては, 階層モデル, ネットワークモデル, 黒板モデルなどが従来から知られており, さらに, 最近では, コネクショニストパーサーなどが提案されている [6]. 本題に入る前に従来のアプローチについて概観する.

2.1 機能分割構成法

自然言語のためのアーキテクチャの最近の研究動向として, 機能分割構成法から能力 (行動) 分割構成法への移行を挙げることができる. 従来の音声言語システム構成法は, システムをより小さい機能のサブシステムに分解し, 個々のサブシステムを実現するとともに, サブシステム間のインターフェースを定めることによって, 全システムを組み上げるという機能分割構成法であった. 言い換えると, 機能分割構成法とは, 大きなシステムを扱いやすくなるまで, サブシステムに分解していく還元主義的なアプローチ (分割統治法) である. 情報は, インターフェースを通じて個々のサブシステム間を流れていくので, 機能分割構成法を, 直列型構成法とも呼ばれる. 機能分割構成法による音声言語システムの概念図を図 1に示す. 図中で枠で囲ったのは, 個々の機能である.

機能分割構成法でシステムを構築しようとすると, 以下のような問題点がある.

- 一部のサブシステムを検査するためには, 全体のシステムが動いていなければならない. (少なくとも全インターフェースは規定されていなければならない.)
- システムを理解するには全システムを知っておく必要があるので, 全システムを理解することが難しい.
- 状況の変化に対してシステムを段階的に対応させることが難しい.

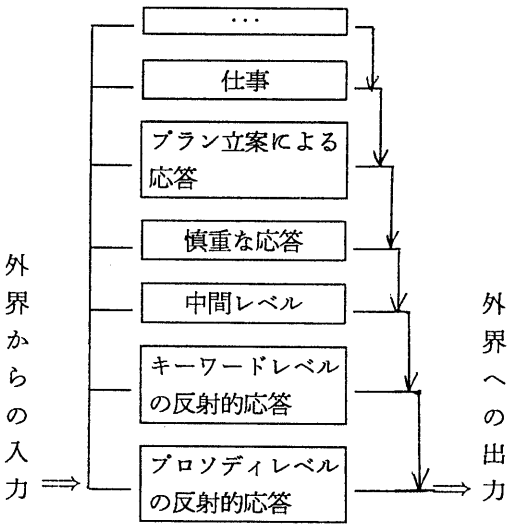

図 2: 能力レベルによる自然言語理解の分割例

2.2 能力分割構成法

機能分割あるいは直列型システム構成法の欠点を克服するために能力分割あるいは並列型システム構成法が提案されている. Brooks は, 行動 (behavior) によって分解する能力分割構成法による自律型移動ロボットの構築を提案している. 能力はレベル分けされており, 能力の上位レベルでの行動が下位レベルでの行動を包摂 (*subsume*) することによって (上位レベルでの行動の決定が下位レベルでの行動の決定よりも優

先する), より高度で知的な行動を実現することを狙っている. この構成法は包摂アーキテクチャ (*subsumption architecture*) と呼ばれる [1]. 実際, 包摂アーキテクチャによって数種類の自律型ロボットが作成されている. 包摂アーキテクチャによる自律型ロボットの行動は, 予め定められたものではなく, 場面, パラメータなどによってその行動が創発 (*emergent*) されるのである. (以下, 能力とその結果現われる行動とを厳密に区別せずに使用する場合があるが, 文脈からどちらを意味するのかは明らかであろう.)

包摂アーキテクチャの特徴をまとめると, 以下の3点になる.

(1) 能力のレベルを積み上げることによって, 外に現われる行動を段階的に知的にできる.

(2) 直接外界とコミュニケーションすることによって, 状況に応じた行動を取ることができる.

(3) 個々の能力がレベルで実現されているので, システムの行動が具現化し, 理解がしやすい.

能力分割構成法による音声言語システムの概念図を図 2に示す. 能力レベルは図中で枠で囲んで示した. 包摂アーキテクチャでは, 各能力レベルは基本的な操作を行う複数のエージェントから構成される. (図中の枠の中は複数のエージェントが入る. 機能分割構成法の場合には必ずしもこのように構成する必要はない.) エージェントは局所的にかつ非同期的に実行され, 相互にはきわめて疎にしか結合していない. また, エージェントの各々は, 外界と直接コミュニケーションを行うことができる. 包摂アーキテクチャでのエージェントは, 機能別分割法による機能モジュールと比べ, ずっと基本的な機能しか有していない. 言い換えると, 包摂アーキテクチャでは, 行動を担当する個々のタスクは, 全体問題を自分が扱える, より単純な部分問題に縮約して, それを解く能力を有していることになる.

3 マルチエージェントシステムとしての能力分割構成法

エージェントの集合であるマルチエージェントシステムが Minsky の「心の社会」[12] のような意味で, 全システムを構成するための能力分割構成法を検討しよう.

独立したタスクを担う自律的なオブジェクトを**エージェント**と呼ぶ. 一つの能力 (行動) は, エージェントの集合である**エージェンシー**で実現される. 全システムはエージェンシーの集合, すなわち, エージェントの集合で構成される. エージェントは入力ポートと出力ポートを持ち, それらは他のエージェントあるいは外界と接続されている.

エージェントあるいはエージェンシーは, システムのゴールとは独立にそれ自身でゴールを持つことができる. したがって, システムには複数のゴールがあり, かつ, それらのゴールがおたがいに矛盾していてもかまわない. エージェントは自分のゴールを達成するために下位エージェントが出す結果を自分が出す結果で置きかえたり (suppress), あるいは, 下位エージェントへの入力を禁止したり (inhibit) できる. たとえば, 反射的な応答を慎重な応答で suppressor すれば, 十分時間があるときには慎重な応答, 時間がないときには反射的な応答, というように切り換えができる.

システム全体のゴールは陽には与えられない. つまり, システム全体のゴールは, システム中の競合するゴールのうち, その時点で最も適切なゴールを選択し, それが規定する行動をとる, というメタレベルでのゴールであると言い換えることができる.

システムはエージェントが相互に結合されたネットワークで表現され, システム総体としての行動はネットワーク上での情報のやりとりによって決ることから, 本稿では, このネットワークを **行動ネットワーク** (*behavior network*) と呼ぶ. では, 行動ネットワークで具体的行動がどのように決まるのかについて見ていこう.

3.1 行動選択ダイナミックス

完璧な人間の行動などあろうか. 様々な場面でいつも**満足のいく** (*good enough*) ような決定ができればよいはずである. 満足のいく行動選択とは, 次のように言い換えることができる [10]:

- ゴール指向の行動が望ましい. とくに, 複数のゴールが同時に達成できればさらによい.

- 現在の状況に相応しい行動が望ましい．とくに，あらゆる可能性を追求し，予知しえないような変化する状況にも対応できればよい．
- 他に理由がない限り，進行中のゴールやプランに寄与できるような行動が望ましい．
- 危機的状況あるいは競合するゴールを避けるために，先読み，プランが立てられること．
- 頑健性があり，システムの一部が機能しなくなってもシステムとしては停止しないこと．
- 反射的であり，迅速であること．

Maes は，上記の課題を達成するために，「心の社会」[12] や包摂アーキテクチャ [1] で示される，心のない能力エージェントの社会で知的システムを構築しようと提案している．

マルチエージェントシステムで具体的な行動を決定するためには，次の 2 点を規定しなければならない．

(1) エージェントの活性化，不活性化の方法．

(2) エージェント間の協調を決定する要因．

従来，いくつかの方法が提案されてきた．たとえば，Brooks はエージェントの制御情報の流れををハンドコードし，ハード的に結線する方法を提案している．分散 AI では，全体の制御を階層的構造で行う組織論的方法がよく使われている．

それに対して，Maes は，行動選択を行うのに次のような方法を提案している．

- システム全体としての満足のいく行動選択は，エージェント間で活性化 / 不活性化を行うことによって実現する．
- 管理エージェントや大域的な制御は不要．

能力エージェントは，前提条件リストを持ち，それがすべて満たされれば実行可能となる．他のエージェントあるいは環境から与えれた入力が，前提条件のどれかを満たすときには，活性化エネルギーを得，エージェントの活性度が上がる．活性度は，実行可能なエージェントのどれを選択するかを決定するのに使用される．エージェントは付加リストと削除リストという 2 種類のリンクを持つ．エージェントが実行されると，付加リストにあるエージェントに対しては正のエネルギーを与え，削除リストにあるエージェントに対しては負のエネルギーを与える．また，活性度には持続時間があり，それを過ぎると徐々に自然崩壊する．

能力エージェントのネットワークによってエージェント集団はお互いに活性化し合ったり，禁止し合ったりすることによって，現在の状況とゴールに対して最良の行動を提示するエージェントに活性化エネルギーが貯えられることになる．活性化エネルギーが閾値を越え，それが実行可能であると，そのようなエージェントが活性化され，実行される．

活性化エネルギーは，外界から与えられ，リンクによって，前提条件が満足されるエージェントへと流れていく．さらに，エージェントのグローバルゴール群からも活性化エネルギーが流れだす．競合するゴール群が自分のゴールを達成しようと活性化エネルギーを流すのである．ゴールには 1 回限りのものと永続的なゴールがある．一旦達成されたグローバルゴールからは，削除リストを使用して逆に禁止することもできる．

活性化エネルギーの拡散は，4 つのグローバルパラメータ — (1) 活性化の閾値．(2) 真である条件ごとにネットワークに注入されるエネルギー量．(3) ゴールがネットワークに注入するエネルギー量．(4) 実行されたゴールがネットワークから削除するエネルギー量．— を使用して調整される．

このシステムをロボットの簡単な仕事 (板にやすりがけをし，自分にペンキを吹きつける) に適用して，ゴール指向性，状況指向性，適合性，頑健性，先読み等の機能が実現されることがシミュレーションによって示されている [9]．さらに，大域的なパラメータよって，ゴール指向対データ指向，慣性対適合性，ゴール競合に対する敏感性，熟慮対速度などのトレードオフを制御できる．

このような計算モデルは，能力がプログラムされているという従来の AI アプローチ，あるいは学習の結果が能力であるというコネクショニスト的アプローチとも異なっている．また，記号システムや部分記号システムからなるハイブリッド型システムでもない．Maes の行動選択ダイナミックスは，記号的な構造表現をコネクショニスト的計算モデルで統合したものであり，両者の利

点を合せ持つ.

我々は, 音声言語システムを構築する計算モデルとして Maes の行動選択ダイナミックスを出発点とした.

3.2 エージェントの記述

エージェントの記述は, Brooks の行動言語 [2] に従っている. 次のマクロにより

```
(defagent 名前
   :input   入力ポートリスト
   :output  出力ポートリスト
   :decls   ローカル変数宣言リスト
   :processes ルール
   :precondition 前提条件リスト
   :activation   活性度
   :threshold    活性化閾値
   :continuance  活性度持続時間
)
```

エージェントを指定する. エージェント間の接続には,

```
(connect  source {行き先}...)
```

を使用する. ただし, ネットワークでの削除・禁止を指定するには, 行き先として以下のものを指定すればよい.

```
((suppress 入力ポート))
((inhibit 入力ポート))
```

4 行動ネットワークによる実現

エージェントがinhibitor と suppressor で結合された行動ネットワークとして音声言語システムを実現する.

4.1 単語認識レベル

最も単純な行動レベルは, 単語を認識し, 最も可能性の高い単語を順次出力していく. たとえば, HMM (Hidden Markov Model) 法によって, 予め与えられたコーパスをネットワークに変換しておき, 音声信号から単語や単語列を抽出する. また, ワードスッポティングにより特定の単語を同定することも可能である. このレベルは, 図3に示したネットワークのの一番下の部分である.

4.2 反射的行動レベル

音声認識と結びついた行動様式の一つは, 反射的行動であろう. たとえば, 大声でおこられた場合にはすぐに「すいません」と謝る, というのは反射的行動の例である. 反射的行動レベルを行動ネットワークで示したのが図3である. 図で bold 体で示したのは入力と出力を示す.

音声信号からピッチ, パワーなどを抽出したプロソディ情報と, 身振, 手振, 顔つき, ウィンクなどの非言語的な情報から抽出したプロソディ情報とから, プロソディの情報の重みを計算する. もし, 閾値を越えている場合には, 最下位からの単語の出力を一定時間禁止する. これによって, プロソディ情報から重要と思われるような単語をきわ立たせることができる. もし, 重みが閾値を越えなければ, 最下位からの単語の出力はそのまま継続する.

プロソディとしては, 音声信号の強勢, 持続時間, 抑揚 (ピッチ) などがある. これらの情報を生データとして扱うだけでなく, 情報を記号化する必要があろう. 記号化にあたっては, Clynes の8つのカテゴリーが役に立つと考えられる. Clynes は, 音声信号の急激な変化のパターンを8つのカテゴリー — 愛情 (love), 嫌悪 (hate), 悲嘆 (grief), 歓喜 (joy), 尊敬 (reverence), 立腹 (anger), sex, 無感情 (no emotion)) — に分類している [4, Figure4]. このカテゴリー分けは, 音声信号だけに限らず, 視覚的な変化, 身振などにも共通しているという. 強勢, 持続時間, 抑揚などをプロソディの1次特徴と位置付ければ, プロソディーの2次特徴としてこの7つのパターンを使用することができ, 発話意図を解析するのにも利用できると期待される.

4.3 並列・並行計算の可能性

音声認識システムがデータベースシステムとのインターフェースとして使用されているとしよう. このような場合には, 逐次出力される単語を参考にして先行計算 (*speculative computation*) が可能となる. 最初に単語 Which が出力され, 次に language が続けば, この発話は言語についての話題であることが分る. その時点で language

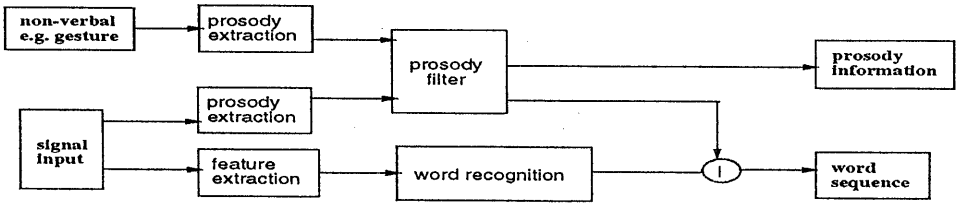

図 3: 反射的行動レベルのネットワーク

についてのデータベース問合せを始めてみよう. もし, このゲスが当たった場合には, 極めてすばやい応答をすることできる. もちろん, この計算はすべての入力処理が終わるのを待たずに行っているので, 間違ったゲスである可能性は高い. しかし, マルチプロセッサ環境の下でプロセッサに余裕がある場合には, この種の先行計算をやってみる価値は十分にある*.

4.4 上位レベルの処理

岡田が提案している音声言語システム [14] をもとにしたシステムの概念図を図4に示す.

単語認識よりも上位レベルにある処理は, キーワードあるいはテンプレートだけで発話を処理する能力レベルである. 聞き流しているときに, 突如聞き耳を立てるという聞く態度の変化は, 図中の template or keyword というエージェントからの出力が上位レベルに接続されているリンクによって生じる.

また, 上位レベルでも同じような「聞き流したり」, 「聞き耳を立てたり」する現象が生じる. たとえば, Unification-based grammar のなかでも, 話されている内容に新しいものがないと分った時点で「聞き流す」ことができるし., また, 発話のトッピクスが分った時点で改めて「聞き耳を立てる」ことにもなる.

トリガーをかける方法には活性度を使用する方法と, inhibitor を使用するのと 2 通りある. 図4に示したのは前者である. 活性度を使用するのは, 3.1 節で説明した. 活性度が自然崩壊すると注意が散漫になり「聞き流す」ことになり, トリガーとして活性エネルギーが与えられると, 活性度が増し, それが閾値を越えると「聞き耳を立てる」ことになる. 活性度を使用すると, 「聞き耳を立てたり」「聞き流したり」する現象が, システム内部では滑らかに創発する.

もう一方の方法では, 言語処理を担当するレベルへの入力を inhibor 経由にし, トリガーをかける側の出力を inhibitor のコントロールに接続する. 通常は, inhibitor を on にしておくことによって, 入力が入らないように, すなわち, 「聞こえない」ようにする. トリガーをかけるときには, コントロールを落とし, 入力が inhibitor を通るようにする. inhibitor を使用する方法では, システム内部では「聞こえない」「聞こえる」というように劇的な変化が生ずる.

このように, 活性度を用いた方法はコネクショニスト的なアプローチであるのに対して, inhibitor を用いた方法はデジタル的なアプローチとなる.

4.5 多重文脈による構文・意味解析

行動ネットワークはマルチエージェントシステムであり, シングルプロセッサ上のマルチプログラミングというよりは, マルチプロセッサシス

*従来の提案されてきたアルゴリズムでは, アルゴリズムが実行される時点ですべてのデータが揃っているが仮定されてきた. このようなアルゴリズムは, off-line アルゴリズム と呼ばれている. たとえば, quick sort を初めとする sorting の多くのアルゴリズムは off-line アルゴリズムである. それに対して, すべてのデータが揃っていなくてもアルゴリズムの実行を開始するようなものは, on-line アルゴリズム と呼ばれる. 上述したような先行的計算には, on-line アルゴリズムが極めて有効と考えられる. たとえば, insertion sort. データベース問合せでは, 問合せ最適化を on the fly でやるようなことも検討する価値があろう.

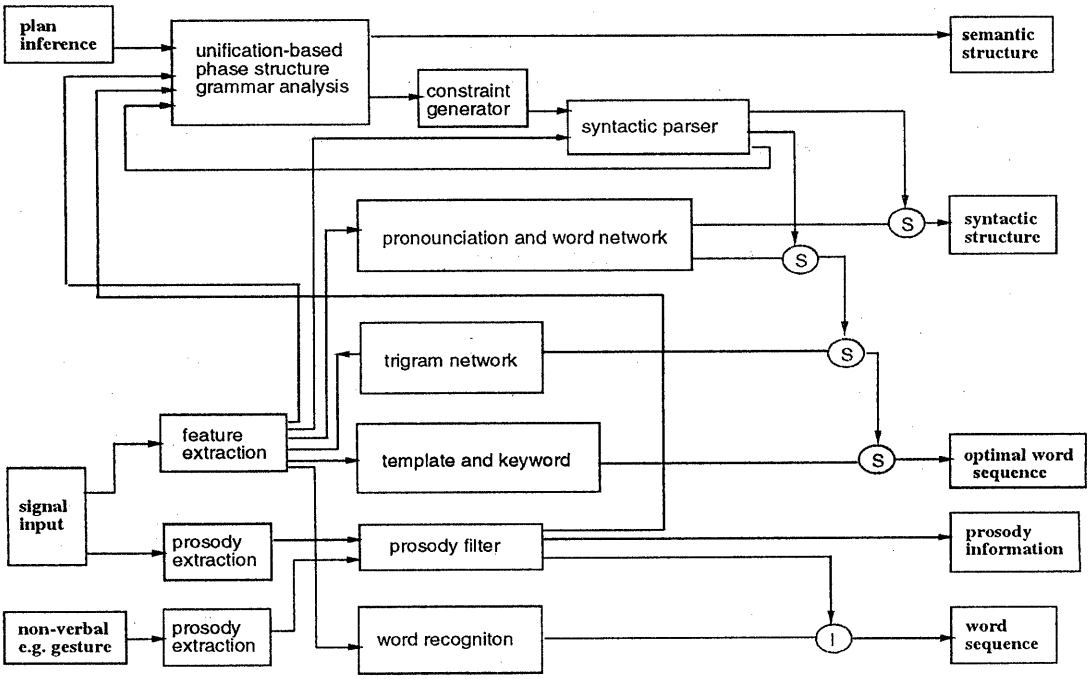

図 4: 行動ネットワークよる自然言語理解

テム上で実行することを想定している．複数のプロセッサを利用して，様々なレベルで並行・並列処理 (concurrent and parallel processing) を行うことによって，処理の高速化・高度化が狙える．さらに，能力分割による複数レベルの並行処理だけでなく，各能力レベルでの信号処理，構文解析，意味解析で複数の候補を同時に追求する多重文脈処理 [19, 21] もエージェントを置き換えることで行動ネットワークに組み込める．

5 考察

5.1 マルチエージェントシステムの全体的挙動 — カオス・プログラミング

システム全体の行動がマルチエージェントシステムの創発的な計算で決るという性質からは，似かよった初期状態であっても，パラメータの設定等がすこし異なるだけで全く異なる行動が生じる可能性が生ずる [5, 20]．マルチエージェントシステムでは，システム全体があるゴールに向かって収束するという従来のアプローチよりも，少し違った状態からでも全く違った行動が生じる可能性があるという，カオスのほうが自然である．本稿でのマルチエージェントシステムの構成法が実際に，システムの行動が初期値に敏感であり，予測ができないという意味でのカオス・プログラミングになっていることを確認することは，今後の課題である．

5.2 マルチエージェントシステムでの文法

本稿で論じたシステムの文法は，Chomsky が提案する universal grammar のような固定した文法ではなく，変化，あるいは進化する文法を考えている．各能力レベルでその処理に応じた文法を用意することを仮定した．レベル毎に文法が用意されているので，必要最小限の機能しか含まず，その処理は効率がよいものとなる．しかし，新たに能力レベルを追加するときには必ずそのレベルに応じた文法を，その都度用意しなければならない．

本稿では検討しなかったが, 全システムの文法を統合するアプローチがある. すべてを一括して記述する単一化文法(unification grammar) を一種類だけ用意する. そして, レベルに応じてその文法のうちの不必要な機能をマスクしてしまうのである. レベルに応じたマスク機構は, (1) レベル記述を単一化文法の制約条件として組み込む, あるいは, (2) レベルを記述する制約条件で単一化文法をプレコンパイル, あるいは, on-the-fly でコンパイルして, コンパクト化する, といった方法が考えられる. これらのマスク機構は, 岡田 [14] で採用されている. 単一化文法による文法の統合化についても今後の課題である.

5.3 音声レベルでの取り扱い

本稿での議論では, 音声レベルでのアダプティブな処理については触れてこなかった. むしろ, 音声レベルでの特徴抽出という処理は 5ms 毎にフレーム分けされ, それが HMM 等の単語同定ルーティンに与えられるとしてきた. しかし, 聞き耳を立てたるという行為は, 上位レベルだけでなく, 音声レベルでも取り込まれるべきである. たとえば, 大勢の人が話している中で特定の個人の話し声にだけ注目したり, 騒音下で注意して聞いたりするという行為が, 音声レベルで現われなければならない.

聞き流す, 聞き耳を立てることの音声信号レベルでの取り扱いは, 稿を改めて報告する.

5.4 発話生成への応用

これまでに提示してきたのは, 音声言語理解であった. しかし本稿での計算モデルは, 音声生成にも応用できる.

「口ごもる」「言い淀む」「言い直す」といった発話現象が生じプロセスとして, Levelt は自己モニタ機能による現象であると説明している [7]. 自己モニター機能とは, 発声者自身が自分の発話の聞き手となり, その発話が適切であるかをチェックする機能である. この自己モニター機能は, 一種の音声言語理解システムであり, 実際の発話 (overt speech) だけでなく, 発話意図を実際の発声に落す過程で作成されるに内部発話 (internal speech) に対しても働く.

マルチエージェントシステムの言葉で言い換えると, 次のようになる. 自己モニタからの情報が発話生成エージェンシにフィードバックされ, 割込みエージェントが活性化される. さらに, 活性度が閾値を越えると, 割込みエージェントが実行され, 発話の修正が行われる. ここでのポイントは, 活性度を考慮したフィードバックであり, Levelt の枠組にはない. このような機能は, 本稿で提案した音声言語理解システムを自己モニタという新たな能力レベルとして追加すれば実現できると考える. 最後に, 音声言語の生成に対する我々の立場を以下にまとめる [18].

> 「口ごもる」「言い淀む」「言い直す」といった発話現象を, 言語生成および発話生成の状況, 条件によって生じた創発的な現象ととらえる.

6 おわりに

本研究での我々の立場は,

> 人間の言語活動は, さまざまなゴールを有したエージェントの集合から構成される社会が行う状況によって決る創発的計算 (emergent computation) として捉える.

このような創発的計算は, 競合するゴールの選択過程, あるいは, 複数のゴールを同時に満たす現象と外部には現われる. これは何かシステム全体を制御するものがあり, そのコントロールの下で動いているようにも見える. いわば, 「メンタル OS」 [22] の存在を仮定してもおかしくない. しかし, 我々はそのような制御ルーティンではなく, 自律的なエージェントで構成される社会の創発性に注目した.

また, 従来の研究では, 両極端な行動のトレードオフによってシステムの具体的な行動が決るという考えが強かった. たとえば, Newell は, 両極端なシステムとして, 熟考システム (deliberate system) と経験型システム (preparatory system) とをあげている [13]. 熟考システムは知識探索

によって, 問題を解析し応答を生成するのに対して, 経験型システムでは, 貯えてある経験を探索し, 引き出された経験を現在の問題に適合させることによって, 応答を反射的に生成する. 両者のトレードオフは, どれだけの計算資源が割り当てられるかで決ると主張している.

本稿では, 音声言語の様々な現象はそれぞれが目的を持ったエージェントの集合であるマルチエージェントシステムの状況や条件に影響を受ける創発的な現象である, という立脚点の下に, 音声言語理解の計算モデルを提案した. 今後, 具体的なシステムを実装していき, 「聞き流したり, 聞き耳を立てたりする」コンピュータを実現する. また, このようなアプローチを音声合成システムにも適用し, 様々な音声言語活動が統一的に取り扱うことができることを実証していく予定である.

我々の立場は, 現在主流である認識率の向上を狙う研究とは異なっており, 適当なところは聞き流し, 重要なところははっきり聞くというような機能が音声研究・自然言語研究の両者を統合するときのキーアイデアではないかと考える. しかし, 認識率をその絶対的な評価尺度としないことにためにいわゆる工学的なセンスでのアプローチのよし悪しを計る尺度がないという問題を抱えている. これは, ヒューマンインターフェース研究全般が抱える大きな問題でもある. 今後, 評価方法についても検討していきたい.

なお, 本稿で提案するアーキテクチャは, 音声言語の分野に限らず, マルチエージェントに基づく人工知能分野一般に適用できる. 応用分野の拡大についても今後検討していく.

最後に, 日頃ご指導をいただく NTT 基礎研究所情報科学部 竹内郁雄リーダ, ご討論いただいた竹内グループの島津明主幹員を中心とする同僚, ヒューマンインターフェース研究所 大里延康主幹研究員, (財) 計量計画研究所 大塚裕子さんに感謝いたします.

参考文献

[1] Brooks, R.A.: A Robust Layered Control System for a Mobile Robot, *IEEE J. Robotics and Automation, RA-2* '86.

[2] Brooks, R.A.: The Behavior Language; User's Guide, MIT A.I. Memo 1227, April, 1990.

[3] Brooks, R.A.: Intelligence without Reason. *Proc. of IJCAI-91*, Sydney, 1991, 569–595.

[4] Clynes, Manfred: *SENTICS, The Touch of the Emotions.* AVERY Publising Group Inc., 1989.

[5] Forrest, S. (*ed.*): *Emergent Computation.* special issue of *Physica D,* The MIT Press/North-Holland, 1991.

[6] 橋田浩一, 竹沢寿幸: チュートリアル「自然言語処理における統合の諸相」. コンピュータソフトウェア, Vol.8, No.6 (1991), pp.3–16.

[7] Levelt, Willem, J.M.: *Speaking: From Intention to Articulation*, The MIT Press, 1989.

[8] 北野宏明: 超並列人工知能の現状と展望. JSPP '92 論文集, 情処学会, 1992.

[9] Maes, P.: How to do the Right Things. *Connection Science*, Vol.1, No.3, 1989, 291–323.

[10] Maes, P.: Situated Agents Can Have Goals. *Robot and Autonomous Systems* '90. *also in* [11].

[11] Maes, P. (ed.): *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, special issue of *Robot and Autonomous Systems*, The MIT Press/Elsevier, 1991.

[12] Minsky, M.: *Society of Minds.* Simon & Schuster, Inc., 1986. 安西訳「心の社会」, 産業図書, 1990.

[13] Newell, A.: *Unified Theory of Cognition*, Harvard University Press, 1990.

[14] 岡田 美智男: 音声言語のバージングにおける最適な単語列の探索について. 自然言語処理研究会, 情処学会, 1991.

[15] 岡田 美智男: 音声言語システムの研究動向と今後の課題. 日本音響学会誌, 48 巻 1 号, 1992.

[16] 岡田 美智男: 聞き耳を立てるコンピュータ. *in* [23].

[17] 岡田 美智男: 音声言語のバージングとその基本的な処理単位について. SIG-SLUD-9201-2, 言語・音声理解と対話処理研究会, 人工知能学会, 1992.

[18] 岡田 美智男: 「口ごもるコンピュータ」の実現に向けて. SIG-SLUD-9201-11, *ibid.*

[19] 奥乃, 内藤, 岡田, 島津, 小暮: 多重文脈を保持する構文解析・意味解析の統一的手法. 人工知能学会第 5 回全国大会, 11-6, 1991.

[20] 奥乃, 大里: 行動ベースエージェントによるマルチエージェントシステム. Comp 91-68, SS 91-25, コンピューテーション研究会, 信学会, 1991.

[21] Shimazu, A.: Japanese Sentence Analysis as Argumentation. In *Proc. COLING-90*, Aug. 1990.

[22] 島津 明: メンタル OS. *in* [23].

[23] 竹内郁雄編: 「AI 奇想曲」. NTT 出版 1992.