

1. はじめに

数学的な理論は、物理的に可能な時間、空間、オブジェクトの構成を提示するものではありません。集合論の操作も実時間で動作するものとするには、理論を更に技術へと結び付けていかねばなりません。「鉄腕アトムの設計書」は基本的に数学的な理論の世界のものです。今回は、これを物理的な可能な技術へと落とし込むことを考察していくつもりです。

電子頭脳はロボットに埋め込んで知的作業を実現する装置です。電子頭脳への入力ロボットのセンサー情報であり、電子頭脳からロボットに出力する情報はアクチュエータ情報です。基本的に、電子頭脳とロボットの結合を定義する機構がなければならないという事実はこの点にあります。

センサーからの情報によって、電子頭脳の回路とプロセスは発火を開始し、電子頭脳の持つ行動（アクチュエータ情報の制御下にある）の成長・制御機構の配下で動く創りつけの機構によって発火連鎖は処理され、管理され、最後にアクチュエータの情報としてロボットに与えられるものです。

本アーキテクチャの特徴は、人工知能の基盤システムというものを、自律学習していくものとして実現する一方で、人手によって構築する部分空間を基盤システムに差し込んでいくということを可能とするところにあります。システムの整合性をパターンマッチング機構を中心にして基盤システムが保証していきます。

本アーキテクチャはまた、センサー値やアクチュエータ値の (key,value) を要素ノードとして、プロセスやデータエリアの集まりである領野が構成され、目的をもって構成される要素ノードや領野の集まりである空間からなるものです。

要素ノード、領野、空間はノードとして番号を振られて管理されます。この中でも特に領野という固まりが重要になります。学習による能力の獲得も、超平行処理も、パターンマッチング、評価処理も、この領野という単位で隠蔽して実現されます。大局の指示（空間へのコマンド発行など）が、その空間の属する評価システムによって（評価システム領野によって）適切に微細な具体的処理に落とされ（微細なプロセス領野が選択され）、実行されます。それは、領野の超平行処理を利用することから保証されるものです。

なお、空間は単なるノードの集まりではありません。構造をもったノードの集まりです。領野も構造を持ったノードの集まりですが、作りつけのデータエリアやプロセスを隠ぺいする点は空間と異なります。空間は、目的を持って汎用人工知能が作り上げるノードの集まりで、世界座標系とか、身体座標系とかが相当します。

上位ノードは下位ノード群をアンサンブルと見て、平行処理と互いの調停処理を期待しています。

2. 核となる機構

電子頭脳に作り込むべき核となる機構は次のものです。これらは、基盤領野をそれぞれ形成します。

- (1) セット管理・操作機構
- (2) 発火機構
- (3) 記号・コマンド生成機構
- (4) パターン発見機構
- (5) パターンマッチング機構
- (6) 相同性・相違性判別機構
- (7) 領野分散処理機構
- (8) 数学的能力

これらと、その展開であるいくつかのシステムを以下に説明します。

2. 1 セット管理・操作機構

パターンの発見はデータの重ね合わせの解析によって得るセット群を構築することです。また、このセット群はパターンマッチングに役立つ配列のセットを作りこむものです。セットができた後の操作を駆使することによって、さまざまな人工知能の能力は実現されるのです。

セットの管理は、リストで行い、リストの要素は整数で、昇順に並べられたものとなっていることが高速なセット処理の実現に必要です。そもそも、センサー値のセットが基本的に整数（センサーの識別子が整数）の昇順になっていますから、その部分セットとして全てのセットがなっていますから、整数の昇順であることという制約は負担ではありません。

2.2 発火機構（グラフなぞりプロセス群）

プロセスが起動するのは、プロセスが属すセットへの発火指示によります。自律的プロセスである意志システムの起動もセンサーからの入力を発火の起点になるはずで

す。発火システムが爆発しないのは、超平行処理を人工知能の動作の基本にしているからです。何かを意識した処理をしていますが、多くは、無意識下で動作し、その無意識下で完結していくものなのです。

2. 3 記号・コマンド生成機構

発火したプロセスやセンサー値のセットは番号が振られます。その番号を生成し、相同性と相違性からカテゴリを作る場合に新しい番号を振るのも、その記号・コマンド生成機構（意志システム）が行います。コマンドの生成にはセット処理と公差法アルゴリズムが利用されます。

コマンドには単純な指示を発行するものと、コマンドパターンを提示し、その中の変数を検索で得る（prolog のユニフォーメーション機能に相当）ものがあります。

2. 4 パターン発見

センサー値を配列で表わし、その重ね合わせの状態を蓄積し、重ね合わせの濃淡の共起性を解析することで、パターンを発見します。発見したパターンには記号（番号）を振って管理します。

パターン間の関係（これもパターンです）も単語やコマンドとして収集していきます。

シミュレーションによる事態の推移を予測するとか、モンテカルロ法で、試行錯誤していくとか、こうした予測能力は、パターン発見の中で、パターンがグラフ構造を構成していく中で得ていく能力です。シミュレーションはグラフ構造のデータを評価しながらなぞっていくプロセスで、モンテカルロ法は飛び飛びに在るグラフ構造の島を、或確率でジャンプしながらなぞっていき結果を得ていくプロセスです。

2. 5 パターンマッチング機構

センサー値、アクチュエータ値共に、識別子と値という対になっています。いわゆる (key,value) の形式で管理されます。Value を配列に展開して、その配列の要素で明示するようにすると、(key,value)間の関係が順序という関係で明示されるようになります。このことによりパターンマッチング処理を実現します。

次のようなパターンマッチングの問題を解くには、言語をイメージ空間に投影して、そこで、イメージでマッチングすることが重要かと思います。

(例)

知識：バスに乗って長野駅の近くの本屋さんに行った。

マッチ項目：「街にいった」は正しいか？

マッチ項目：「買い物に街に出た」は正しいか？

これらは、言語の世界でマッチングしようとするると困難を生じる例です。

2. 6 相同性と相違性の判別機構

パターンが一致するものを相同とし、一致しないものを、その一致しない度合によって強い相違性を持たせて表現し管理します。それは単語としてまとめられ、コマンドとして表現します。

相同性と相違性を判定する仕組みは、2つのノードへと接地することで行います。1つのノードに集約する（パターンが同じ）ものは相同であり、2つのノードに分かれて集約するもの（パターンが異なるもの）を相違とし、その2つのノードの距離を測定することで、相違性の強さとして定義します。

2. 7 領野分散処理機構

データエリアの同じタイプのもの集まりとか、同じタイプのプロセス群とかをまとめた空間（セット）を領野とします。データエリア固有のプロセス群では、領野を単位とした分散処理をします。

- ・ステートマシン
- ・プロダクションシステム
- ・学習システム
- ・解析システム
- ・評価システム
- ・直観システム
- ・調停場

などがあります。

データエリアの基盤は次のものです。

- ・マップ・・・・・・イメージを表現します。
- ・セット・・・・・・全ての事象はセットを構成します。
- ・グラフ・・・・・・ステートマシンやプロダクションシステムを構成します。

事象はノードとなりますが、そのノードはセンサー値とかアクチュエータ値とか、解析によって得るカテゴリの識別子のどこかにセットとして、まずは識別子が与えられます。そのようなセットが体験を繰り返す内に、セットの積集合として精密な識別子が与えられるようになります。公差法による相同性・相違性弁別のアルゴリズムの実現がこれです。

2. 8 数学能力

一対一対応とか、相同性・相違性弁別、数直線とか、順序・加減とかの人工知能としての基本能力を作りこんで、公開します。基本的に数直線の内部ノード列が数学能力の接地を実現します。それは、時間と3次元空間の基本にもなります。

2. 9 セット処理が支える人工知能の機能

ブール代数などのセットの処理は、人工知能の基本の能力であり、ここでは、その展開の様子を記述したいと思います。

セット処理で、ノードを管理します。事象だけでなく、領野、空間とも、ノードとして扱います。セットは入れ子構造になります。そのため、単純な統一したシステムを実現することができます。このアーキテクチャの要のデータエリアがセットであるとする所以です。

セットの相同性と相異性を判定して、基盤システムにどのようにこのセットとセットに付随するプロセス群を組み込むかが決定されます。

セット処理を高速に行うために、ノードは識別子の昇順のリストで管理します。もともと、センサー値とアクチュエータ値は昇順の識別子が割り振られていますから、そこから生まれるコンセプトとしてのノードを識別子の昇順にセットに入れることは負荷になりません。

2. 10 評価システム

人工知能の展開である、思考システムでは評価システムが常に広範囲に関わって来ます。何かプロセスが動くごとに、その結果を評価することになります。評価項目も沢山あり、超平行処理で実現されます。その評価システムでもセット処理は基本的に重要です。評価システム領野の入力はデータのセットであり、評価項目は多数のノードの発火として得られます。

評価は構造化されたノードセットに対して行われます。基本的にコマンドとして表現されますが、そのコマンドは一方でイメージに接地している必要があります。パターンマッチングが行われるイメージに接地しているということです。格、オブジェクトがイメージとして提示されて、評価されるのです。評価はパターンマッチングが基本処理なのです。

2. 1 1 学習システム

学習を自律的に行う領野であります。それは、

- ・交差法
- ・強化学習
- ・ディープラーニング

であり、評価システムと同じく、広範囲の機能のなかで利用されます。そうして、それらは同じタイプものが学習空間をなしています。特に、出現の記録の重ね合わせを管理し、パターン発見の基盤となります。そのため、強化学習空間は、全ての事象間の共起に対して情報収集のパスを通してします。

強化学習はイメージの重ね合わせ、チャンク（単語とかの記号列）のネットワークの重ね合わせで得られる情報です。このため、学習システムの基盤となる過程であるといえます。

2. 1 2 解析システム

センサー値を解析するシステムです。また、言語記号からセンサー値を生成するシステムです。

- ・動画像・静止画像解析
- ・音声解析・フーリエ変換
- ・皮膚感覚解析
- ・身体解析

があります。

これらは、身体と外界のイメージを形成しています。基本的にサイバー空間に、時間と3次元空間のイメージノードを形成していくのです。

解析システムはイメージとして重ね合わされ、パターン発見の基盤データとして蓄積されます。重ね合わせは記号によって管理され、その記号のチャンクはネットワークを構成します。解析結果はイメージというマップとネットワークというグラフ構造を為すのです。時間的シーケンス、空間的シーケンス、繰り返しパターンという基本グラフを為します。時間変化は周波数解析というフーリエ解析も重要になります。

重ね合わせの基本として、記号化されますが、記号のイメージが線分であることもあり、長さ、大きさ、曲率という属性が生じます。また、グラフは枝分かれとか、閉曲線とか開曲線とかの属性というか特異点が生じます。メディア毎に特別な解析プロセスが必要な所입니다。ここは、ユーザ空間へのプログラミングを行うことで対応します。

2. 13 直観システム

パターンマッチングを待つセットからアクションセットを連想し、結果を調停場に保存するというのを自動的に行うシステムです。アクションとして、次の物があります。

- (1) シミュレーション
- (2) モンテカルロ法の展開

いずれも各ステップごとに評価プロセスが噛まされていて、有意義な成果のみを調停場に置くように設計されます。

直観領野の入り口はパターンのセットです。パターンマッチングが行われ、マッチしたパターンに対して特定のプロセスが実行されます。そうして結果が調停場におかれるというプロダクションシステムを構成します。

2. 14 調停場

調停場は、各種の成果を保存するセットで、そこには評価プロセス群（領野群）が連想されています。評価結果は所定のアクション（領野）を発火するようになっています。これは学習によって構築される場なのです。

評価プロセスはパターンマッチングシステムですから、パターンマッチングとプロセス選択とプロセス実行と結果集積と集積結果評価がプロダクションシステムを構成していますから、調停場（セットで表現されます）にはプロダクションシステムが添付されているものとして定式化されます。

このように調停場は大きな構造のシステムですので、使いまわしをすることが重要です。調停場がいくつも集まって一つのプロダクションシステムに繋げるという工夫も有効です。

3. 空間

分散処理とか超平行処理機構の構築は、それぞれ目的を持った情報の固まりとして持つべきでしょう。その固まりを空間と呼ぶことにします。空間は多数の領野の組み合わせで実現される上位領野です。空間として与えるべきものは次の通りです。

- (1) 3次元空間と時間空間をもち多くの作業を行う作業空間と、作業空間の中から有用な情報のみを永久記憶に保持する永続化空間の2層構造
- (2) イメージ（センサー情報）を保持するサイバー空間
- (3) コマンドとかパターンマッチングを高速に行うための仕組みであるデータエリアの固まりとしての知識ベース
- (4) データエリアのタイプとかプロセスが同じ部分空間。セットやステートマシン、プロダクションシステムなどがあります。
- (5) 人工知能を構築する人がプログラミングする空間

3. 1 作業空間と永続化空間

作業空間は、人間の脳の海馬に相当する空間で、センサー値、アクチュエータ値、時間、3次元空間という外界をつぶさに反映した空間であり、外界のモデルとなる空間です。永続化空間は、大脳皮質に相当し、コンセプトとして重要と判断した事象、空間を保存する空間であり、生のセンサー値やアクチュエータ値とか、時間、3次元空間情報は捨てられます。

3. 2 サイバー空間

イメージを保存し、利用していく空間です。特に、言語をイメージ化して認知を実現する空間になります。

3. 3 知識ベース

記号、特に言語とかコマンドを保存し、利用していくための空間です。

3. 4 人工知能構成者空間

人工知能を作る人間が AI 言語を用いて構成していく人工知能の部分空間です。人工知能構成者の窓口になる空間です。

ロボットのセンサー（物理センサーと論理センサー）とアクチュエータ（物理アクチュエータと論理アクチュエータ）を人工知能に結び付ける重要なユーザ空間でもあります。オントロジーが設定されます。

4. AI 言語と領野管理

電子頭脳を0から構築するのは大きなシステムなため、現実的ではありません。ある程度、基本的な処理は公開せずに作りこんで、人工知能として持ってほしい意志とか知識を与えたり、電子頭脳から得たいデータを提示するためのコーディングする言語を公開するのが望ましいこととなります。RDBシステムのSQLのようなAI言語の構築を提唱する所以です。

(key,value) データも単語も単語のネットワークもコマンドも領野も空間もノードとして表現されます。人工知能のデータはすべて整数識別子を振られたノードとして表現され、その集まりをセットとし、特に重要な基本的なセットを領野として管理するのもです。ノードの生成は、どれかのセットの要素として創生されるということです。

領野からの出力を集計したものは、調停場に置かれます。Blackbord_case で示す領野が調停場になります。

(例) `make_Set directobject_case Ifrom(1,4,5) in_case j`

[説明] i:作られるセットの名前

j:作られるセットが保存されるセットの名前

(1,4,5): 作られるノードのセットの名前のリスト

AI 言語は主に人工知能構築ユーザのためのものです。ユーザ空間内で使います。特に、ユーザ空間を通して、ロボットのセンサーやアクチュエータを人工知能に接地しますから、センサー、アクチュエータの定義を実現するインタフェースを持っている必要があります。

5. 意志、言語、思考、行動、認知の各機構の実現

基本的に AI 言語で構築する空間として実現していきます。そうして、その空間の基本要素（ノード）はステートマシン、プロダクションシステム、評価システム、学習システムです。

5. 1 意志

次の能力を持ちます。

- (1) 記号化
- (2) プランニング
- (3) システム制御の元締め

意志は、思考・行動と深い関係を持ち、意識下の処理の中核となります。思考を通して行動のプランニングをメモリ（知識ベース）に記録し、生活を通して意味のある行動発現を実現していきます。

5. 2 言語

コマンドとか記号（単語）ネットワークとかは、言語の深層表現を成します。外界との会話で使う単語の並びは表層表現となります。また、ステートマシンを作り、対話や物語の話のシナリオを獲得していきます。いずれも学習によって獲得していく能力です。

言語の獲得は強化学習と公差法によるコマンドの形成と単語ネットワークの形成という学習です。

言語の活用の制御は強化学習と公差法によるシナリオ（ステートマシン）の形成という学習です。

言語の獲得に関しては、コンテンツ「言語と思考システムの詳細設計（2019.06.11）」を参照下さい。

5. 3 思考

ステートマシンで表現される知識と、モザイク状に発火した事象から未発火の事象要素を推定発火し、全体を評価して、問題の設定、解決、行動のプランニングをする機構です。学習によって、ステートマシンと未発火事象要素推定能力を獲得していきます。未発火事象の発見の主な手法は、

- (1) シミュレーション
- (2) モンテカルロ法
- (3) パターンマッチングの組み合わせ管理

意識下の思考は、目標管理思考で、無意識下の直観と対になって働きます。意志の一部と目標管理思考が意識です。思考を図形で表現すると、目標管理思考は曲線の流れであり、直観は各点での接線になっているという例えになります。直観は全ての思考の局面で、先頭していく自動的に動くプロセスです。

一方で、思考はイメージ（時間と3次元空間）のマップデータを中核にして達成される能力です。イメージは対象そのもののデジタル・ツインであり、解析して得る結果を基にデータ処理していくと、無限の解析結果を引きずる必要があり、問題を難しくすることを避けるための手段になります。

思考の発現は、文章生成と同じであり、コマンドのシーケンスとして展開していくものです。そのコマンドシーケンスは、2つの構造から制御を受けます。

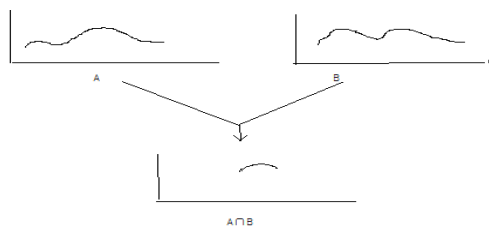
- (1) 個別の試行錯誤の思考の体験を纏めて、事象のネットワークとして記録しておき、そのネットワークを使い、思考を制御していくもの。
- (2) 大きな枠組（例えば、数学の問題を解くとかの）から色々体験を事象群として捉え、交差法によって、大枠の問題群を解く制御チャック群を切り出し構成し、ネットワークを作りアンサンブル法で実際の問題を解いていくというもの。

パターンマッチングがあつて、それを基にしたアクションがあり、そのアクションの結果を評価し、またパターンマッチングから繰り返していきます。プロダクションシステムを構成すると言えます。

5. 4 行動

ステートマシンで表現される知識です。入力意志とセンサー値群であり、出力はアクチュエータ値群です。学習・・・外界を観て真似るとかモンテカルロ法で最適化パスのステートマシンを獲得する手法などがあります・・・によって獲得していきます。

行動制御の構造はステートマシンとして定式化されますが、その構造の大枠のものは、大局的な行動の決定の意思の下に、各運動要素（腕、足、胴体など）のレベルでの姿勢を決定するという意思の表現が制御され、さらに運動要素の下に、筋肉線維のレベルでの伸長、収縮の動きの決定要素の意思（アクチュエータ値の決定）の表現という風に階層をもって実現されていますから、頑健な制御が実現できています。意志とそのときの姿勢の状態とセンサー値は下位の意思（もしくはアクチュエータ値の発生）を決定するステートマシンを構成します。



2つの関数の構成員処理で、共通の関数パターン部分を取り出す

行動制御で使うセンサー値（強さ）はセンサーの並び上に関数を構成します。2つの行動点のセンサー値をAとBとすると、その重なりはAとBの関数値の一致する点群（ある誤差の範囲で一致するということで、ファジー関数の導入が必要になりますが）を抜き出すと、その抜き出した部分関数はAとBの制御点での共通関数ということになります。こうして、多くの制御点の関数群から、部分関数群をチャンク（単語）として切り出すことができます。その単語は行動制御ではネットワークを構成することになります。ステートマシンの入力単語になります。これに応じて出力単語（アクチュエータ値セット）が決まります。

意志は骨格の配置と運動速度の変化（開始点と目標点）を指定しますが、その変化の指定には階層があります。長野駅の端から端まで歩くという意思は、今ある足の位置速度0から次の目標の1歩先の足の位置速度10とかに落とされて、身体の運動という具体的な行動の意思にブレークダウンされます。さらに、関連して腕の位置と速度の変化という意思が平行して設定されます。胴体についてもそうです。さらに、足のこの筋肉をこれだけの強く収縮するというような意志にブレークダウンされるでしょう。その最微細な指示は、こうゆうセンサーの時にこのような筋繊維にこれだけの値を出力するという連想関係に落としこまれるでしょう。これらは階層型ステートマシンを構成するでしょう。しかし、きっちりした明確なステートマシンでは表現できない曖昧性のあるシステムになっているは

ずです。いくつかの近似のステートマシンがアンサンブルとして協調して目的の行動が発現するのです。そのようなステートマシンを構成せねばなりません。

外界の状況というものを世界座標系と身体マップで、身体と関連事項の関係を把握できる情報と、センサー値の状況から上記のステートマシンによって意志が発現することになるのですから、そのステートマシンの獲得は、強化学習と公差法によるしかありません。最微細な指示として情報は収束します。その収束の各段階に意志記号が発生してその記号が担う情報と下位ステートマシンの対応を学習していくのですが、いろいろな大局意志（同じ大局意志でも細部は異なる身体運動になるかもしれません）の中で動く身体の状況を公差法で分割して、最終的なステートマシンへと収束させていきます。

もう一つの行動制御の獲得の方法は、試行錯誤を繰り返し、意志と身体の動きの共起を記録していき、ネットワークとして制御データ（ステートマシン）を構成していく方法です。実際の行動の発現は、意志のシーケンスをプランニングして、目的の行動をネットワークの部分部分を紡いでいくという方法で行うものです。

5. 5 認知

基盤ノードを決めて、これに全ての評価や、感覚（センサー値、アクチュエータ値、論理センサー値）を接地することが認知の基本です。相同性・相違性の判別や数直線などの数学的能力が顕著な例となっています。

プロセスとしては、パターンマッチングが認知の最初のもので、連想推論やシミュレーションやモンテカルロ法による予測の能力を駆使して、状況を把握します。また、イメージのモザイクを組み合わせて創造的な発想も実現し、外界を理解していきます。

認知でとくに大きな能力は、言語の理解です。言語とか画像の認知も知識である構造をもった外界のイメージであるサイバー空間と知識ベースの要素であるコマンド群への対応をつけることであります。それが接地になります。サイバー空間と知識ベースの要素をモデルとして構成し外界を認識するということで、すでに完成されたモデルの各部分に対応させて、認識を実現する場合と、モデルを予測して必要に応じて新しいモデルを作っ部分に対応させていくということを行うことがあります。ただ、モデルを新規に創造するとしてもその組み合わせは既存の部分モデルの整合性を検証した結果ものになっているべきです。嘘を見抜くことも認知の基本能力だからです。

認知では、ダイナミックなモデルの構成とモデル（モデルの部分要素である場合も）と認知対象（言語要素とか画像とか）のパターンマッチングとが基本処理になります。最終的には、センサー（物理センサー群と論理センサー群）、アクチュエータ（物理アクチュエータ群と論理アクチュエータ群）に接地された事象をパターンマッチングの結果発火させることです。認知は更に、シミュレーションによる予測とか、コマンド群を合成してできる事象の再解析の結果とも照合して実現します。認知には、目標を中心とした評価があって、評価結果が求めるものだと、「分かった」という喜びが得られるものです。この喜びが認知の究極の目標になります。

6. 意識下の思考・行動と無意識下の処理

電子頭脳は超平行処理をプロセス実行の基本モードとして提唱してきました。しかし、ただ超平行処理だけで、さまざまな問題が解決するわけではなく、むしろ問題を生じる可能性があります。それは、無秩序な思考・行動の発現につながるのではないかという危惧です。無意識下で様々なプロセスが実行されますが、最終的に意識の下での思考・行動として目的と結果の明確なプロセスが実行されなければなりません。超平行処理に対しては、秩序だった統一した処理の体系をなすソフトウェアアーキテクチャが必要なのです。

緊急事態への警戒は無意識下から意識下への事象の持ち上げを行う最たるものです。この機構は、警戒細胞（ノード）を設けることで実現できます。センサー値がある値の範囲を超えたとき、越えた強さに応じた、警戒注意シグナルを意志システムに送るようになるのです。警戒細胞には各センサー設計時に快適範囲を登録して、警戒細胞は常にこの値を専門にチェックするというアーキテクチャです。

文章は人によって個性がでます。歩く所作でも人によって個性があり、後ろ姿から誰であるか分かるものです。このように、思考や行動には統一したモデルというものがあるようにするのが、まとまった目的をもったものとするために必要なことのように思われます。

思考のちょっとした断片とか、行動のなかの筋肉線維の断片的な動きとか、全体の思考や行動の実現のためには、ローカルな断片プロセスの相互作用が複雑にからまってきます。からまってきますが、全体として統一されているのです。そのため、外乱に対して頑健になります。それは、全体の動きの大枠の動きを定める情報と状況評価項目があって、そのなかに断片プロセスに直接働く情報とか評価プロセスが幾重にも重なって動いていると考えることが自然なことです。これで、目標と結果を明確にもった頑健なシステムが実現していると考えます。その情報と評価プロセス群は電子頭脳が生活するなかで学習によって獲得していくものなのです。

7. オントロジーと事象（カテゴリー）の接地

オントロジー記号（番号）はセンサーとアクチュエータに接地し、意味を定義しなくてはなりません。電子頭脳システムで唯一の意味を持つことがオントロジーの役目です。例えば、「の中に」という意味のオントロジー記号「in_case」は、領域イメージの中の点群であるというような指定の仕方、意味を定義しなくてはなりません。その意味は、電子頭脳全体で同じイメージに接地することが必須になります。

ここで問題になるのは、イメージを描きづらいオントロジーがあることです。「の中に」で例示しますと、円（セット）の中の点にフォーカスを与えて、円の外でなく中側の点群全般たいして、「の中に」ある点であることをどうやって電子頭脳に示すかということです。

全てはセンサーかアクチュエータに接地するという方針の下にこの問題を解決せねばなりません。どうするか。まず、センサーとしてフォーカスというものを設定します。電子頭脳に論理センサーとして主格フォーカスセンサーと目的格フォーカスセンサーとその他の一般フォーカスセンサーを設けて、全ての情報処理にこのフォーカスセンサーを働かせるとするのです。そうすると、主格フォーカスにイメージの内部の点群をあて、外部の点群に一般フォーカスの点群を当てるようにして、「の中に」のオントロジーを定義するのです。

コマンドの動詞で重要となるのは、「be」と「change」と「move」です。「be」は主格の属性をコマンドで定義したり、存在を表したりします。「change」は主格が変化することを示します。「move」は主格が移動することを示します。これらが定義されると「spin」みたいに主格が回転したり、回り道したりすることを表現できます。

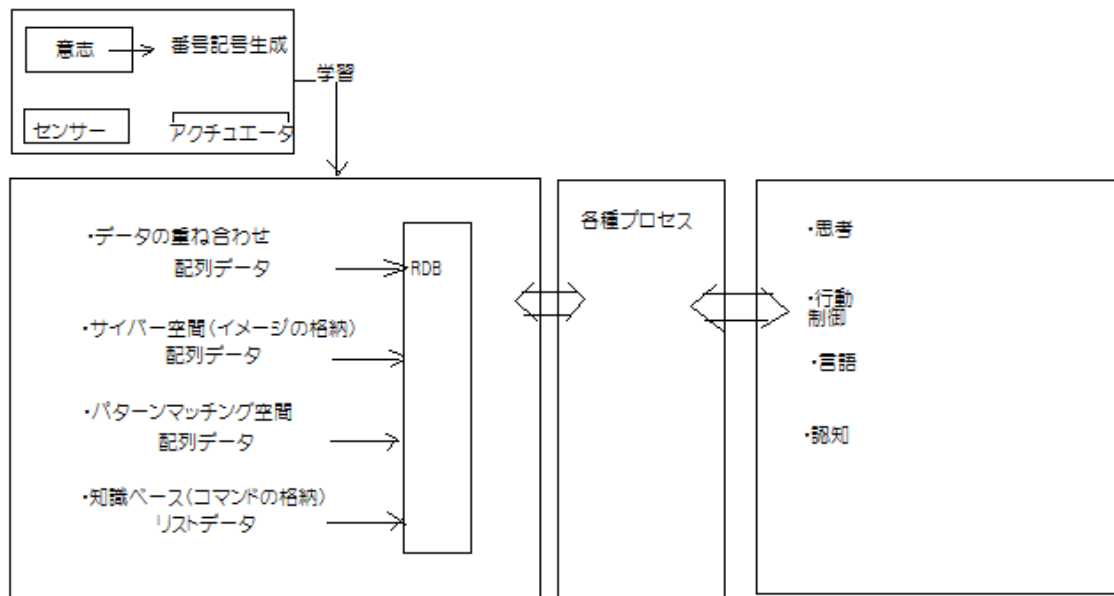
では、「be」は何に接地するのか。一般のフォーカスに接地するとよいですね。とにかくフォーカスの対象になっていることを表現しておけば接地になります。「change」は「時間」と「空間」の中での「事象（オブジェクト）」の変化です。時間と空間は数直線に接地します。そうして事象は点（ノード）に接地します。その点が2つの点に時間的にフォーカスの対応を変えるとイメージに接地すれば良いです。「move」は空間の数直線を時間と共にフォーカスが変化していくというイメージで接地できます。

その他で、論理センサーとして重要なのは、メリット・デメリット・快適と悪・善の差別する電子頭脳創りつけのセンサーです。ロボットの根本センサーであるとして定義すべきものとして作りこむことを推奨します。

8. コーディングの方針

本システムは、自律学習することを示しましたから、自己組織化の能力を備えていることが言えました。また、意志の下にイメージを記号化し、各種処理を自律的に実施していくことが示せたから、自律システムであることも示せました。しかも、その自律システムは、統一的に目的を持って、各部処理が整合性のある行動・思考を実現していることから、意味のある組織的システムであることが保証されるものです。

ここでは、このシステムの大枠の構造をイラストで提示したいと思います。



データは基本的に配列であり、一つの配列が **RDB** の一つのテーブルに対応します。また、リストも、一つのリストは **RDB** の一つのテーブルに対応します。

図に表したパートの枠組みは、本システムの基盤として作りこみます。各種プロセスはユーザが独自のものを作って組み込むことができるようにシステムを構成します。セットとリストとそれら进行操作する **API** が本システムとユーザ空間とのインターフェースになります。

9. おわりに

FPGA や GPU や CPU の領野への割り当てということで、超平行処理を実現することを提唱するものです。電子頭脳は領野のモザイクとなって構成されるものだと考えるものです。

さらに、領野のセットとしての空間の概念を取り入れ、そのいずれも事象と同じノードとして扱うという考え方を実現するアーキテクチャとなるため、基盤プロセス・データエリアの組み合わせを自在に制御でき、自己組織化能力を持つ強力な学習システムを組み込むことを可能としました。

大局のデータの流れを定義すると、その空間の細部では調停場が形成され、そこから更に領野毎の処理が選択的に行われ、そこに細部の調停場が形成され・・・という風に、データ処理の粒度が細かくなって、全体的に適切な処理が実行されるということが保証されます。

セットとしてのプロセス・データエリア管理が強力なアーキテクチャであるといえる局面です。

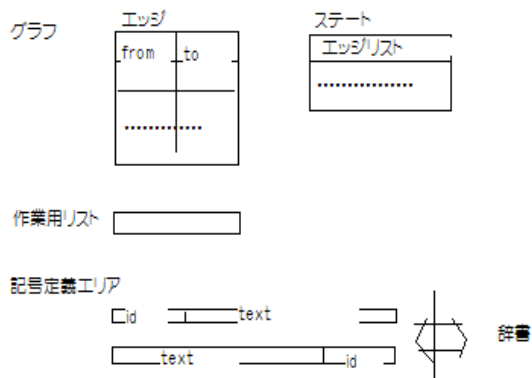
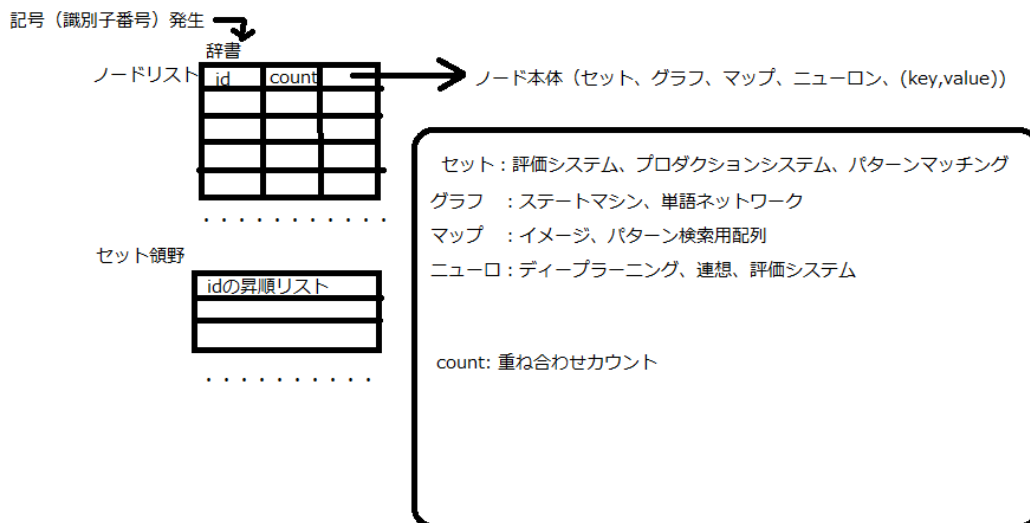
本アーキテクチャでは、セット処理と配列を基としたパターンマッチングとグラフのなぞり処理が重要な核プロセスで、超平行処理で実現するコンポーネントです。あとの処理はローカルな低並列処理で実現できる物です。そして、言語も思考も行動制御も同じ処理であることを見てきました。領野のアンサンブルとして実現できる物だとおうことが言えました。そのアンサンブルは次の2つの方法で獲得されていく物だと言うこともみました。

- (1) 大局から細部へとデータを組織立てていき、チャンクとチャンクのネットワークとして記録していくこと
- (2) 特定のプロセスを試行錯誤で動作させ、それを記録し、記録のネットワークを島状に構成していくこと

結局、ロボットシステムに電子頭脳を埋め込むには、センサーとアクチュエータをユーザ空間で定義し、セットと配列とグラフにデータを埋め込み、電子頭脳とユーザインターフェースを通して、電子頭脳にユーザ空間を組み込むことで行えるということです。

最後に、基本データエリアの構成を図示して記述を終わりたいと思います。

電子頭脳の部品データエリア



記号発生は各種データタイプの違いで、異なって独立に発生させるのが良いでしょう。

おわり