

汎用人工知能プログラミングのモジュール構成を以下の図に示します。

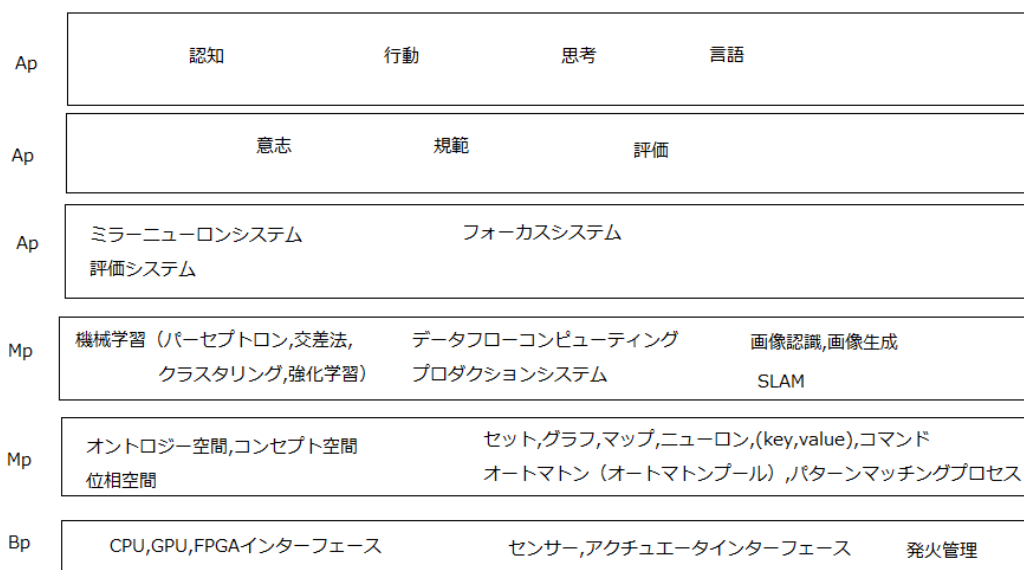


図. モジュール構成

ここで、

- (1) **Bp** は基盤プロセスで、ハードウェアとのインターフェースを担います。センサーやアクチュエータは、汎用人工知能に接続したら、人工知能から自動的にその属性や働きを認知できるような命名規約が成っているとします。
- (2) **Mp** はミドルウェアプロセスで、アプリケーションから汎用的に使われる、データエリアとかプロセス群です。
- (3) **Ap** はアプリケーションプロセスで、人工知能の機能を実現します。
- (4) モジュールはデータエリアでもあり、それらは **AIFactory** というファクトリーパターンによって規定される規約に従って、獲得されます。その中で、必要なプログラムとデータは、CPUにロードされ、実行可能になります。モジュールの呼び出し手続きも **AIFactory** で設定します。
- (5) モジュールは名前によって管理され、その名前はコンセプトの名前です。同一のコンセプトに属するモジュールは、そのコンセプト名を持って管理されることになりま
- (6) モジュールは自分を起動するコマンド表記を定義します。このコマンド表記にマッチしたコマンドが汎用人工知能 (他のモジュール) から発せられると、このモジュールが起動します。
- (7) モジュールは互いに同等な立場で、コマンドを介して通信し合います。フラットな構成を取ります。

## [目次]

### I. モジュール一覧

中心となるモジュールの一覧を示します。コーディングに当たっては更に詳細なモジュールを要素を立て、サブモジュールとして作っていく必要があります。そのサブモジュールも本体モジュール以外から利用できるように作ることが肝要です。

### II. モジュール空間

モジュールの造り方を説明します。動きとか、構成について記述しています。

### III. 知識ベース

学習によって獲得するデータエリアの構成を説明します。

### IV. オントロジーとコンセプト

情報処理の対象要素であるオントロジーとコンセプトの設計について説明します。

### V. タイムスタンプと発火連鎖管理

汎用人工知能の情報処理の基本である発火の連鎖の管理と共起の管理に必要なタイムスタンプを考えます。

### VI. 汎用人工知能の実現性の検証

汎用人工知能として、このコンテンツに記述したことで十分実現できるかということを考えます。

付録1. 行動・思考と意志について

付録2. 会話制御について

付録3. 交差法と言語獲得について

付録4. 多層パーセプトロンネットワークについて

## I. モジュール一覧

AIFactory で獲得する具体的なモジュールの一覧と機能の概要です。

### 1. BpCalculateCell

- cpu,GPU,FPGA インターフェース

- (1) 平行処理の管理
- (2) メモリ管理
- (3) プログラムのローディング
- (4) プログラムの実行

- コーディング

- (1) AIFactory で得たデータエリア (MpSet,MpGraph,MpMap など) を実行するプログラムをシステム領域とデータエリアにロードし、意志システム (ApIntentionSystem) から始まる一連の汎用人工知能の発現を実現する。

### 2. BpSenserCell

- センサーインターフェース

- (1) センサーのシステムへの接続
- (2) センサーからデータを得る

- 説明

- (1) 既知感覚/新規感覚
- (2) ダメージ感覚/メリット感覚
- (3) 自分の一部であるという感覚
- (4) 意味を理解しているという感覚
- (5) 外界の物理状況の各種感覚

- コーディング

- (1) センサーからデータを受けた時、コンセプトを発火したり、コンセプトを作ったりして、自律的に値の連鎖を開始するように設計しておくことが重要です。汎用人工知能システムの方からセンサー値を求めて行くという方法は取りません。

- (2) (key,value)として、データを得ます。

### 3. BpActuaterCell

- アクチュエータインターフェース
  - (1) アクチュエータのシステムへの接続
  - (2) アクチュエータへのデータの送付
  - (3) 例外処理
- 説明
  - (1) 力の強さ
  - (2) 骨格の方向
  - (3) 重力にあらがう力
  - (4) ZMP などの体を支える感覚
- コーディング
  - (1) 対応するコンセプトから特定のアクチュエータに値を発行します。
  - (2) (key,value)として、データを出力します。

### 4. BpNode

- ノード
  - (1) 自分の名前とこれから発する BpEdge 群の各名前を保持
  - (2) 発火管理の対象
- 説明
  - (1) 操作/オブジェクト/属性/格をします。
  - (2) オントロジーとかコンセプトのコアとなります。

### 5. BpEdge

- エッジ
  - (1) BpNode の間をつなぐパス
  - (2) 自分の名前と始点の BpNode の名前と終点の BpNode の名前を保持
- 説明
  - (1) 連想関係を示す
  - (2) 遷移関係を示す

## 6. BpFireManageField

- ・発火管理場

- (1) 発火管理は **BpNode** の活動と休息の 2 状態と活動の強さを制御することです
- (2) ネットワークを構成

- ・説明

(1) 発火管理は、一定の時間間隔ごとにタイムスタンプと **MpConceptSpace** を設けて管理します。

(2) 発火した **BpNode** は **MpConcept** を作り（新規の場合）もしくは、既存の同名の **MpConcept** を利用し、**MpConceptSpace** に登録するということで管理します。

(3) **MpConceptSpace** は **ApForcusSpace** でも管理され、フォーカスの数だけ、**MpConceptSpace** が作られ、フォーカスが当たった発火が管理されます。

## 7. MpOntologySpace

- ・オントロジー空間

- (1) オントロジー名を座標とする位相空間

- ・説明

- (1) オントロジーはセンサーの(key,value)の key であり、value をマップの点（ノード）で表わした物が位相空間になります。オントロジー空間内の点にコンセプトは配置されます(コンセプトの名前を設定する)。コンセプトが衝突したら、更にオントロジー空間を精密な座標で表現したマップとして構築し、前の衝突した点に連想付けます。

- ・コーディング

- (1) センサーの種類毎にオントロジーを命名規約に則って設定する
- (2) アクチュエータ毎にオントロジーを命名規約に則って設定する
- (3) 記号（コンセプト的な）オントロジーを設定する

## 8. MpConceptSpace

- ・コンセプト空間

- (1) コンセプトの名前の集まり。コンセプト名はオントロジー空間の点に置かれます。
- (2) コンセプトはオントロジーの拡張としてオントロジーのように振る舞うものもできます。オントロジー空間を作ることもあります。

## 9. MpConcept

- コンセプト

- (1) BpNode の名前、もしくは BpEdge の名前を保持
- (2) モジュール名を保持

## 10. MpTopologySpace

- トポロジー空間

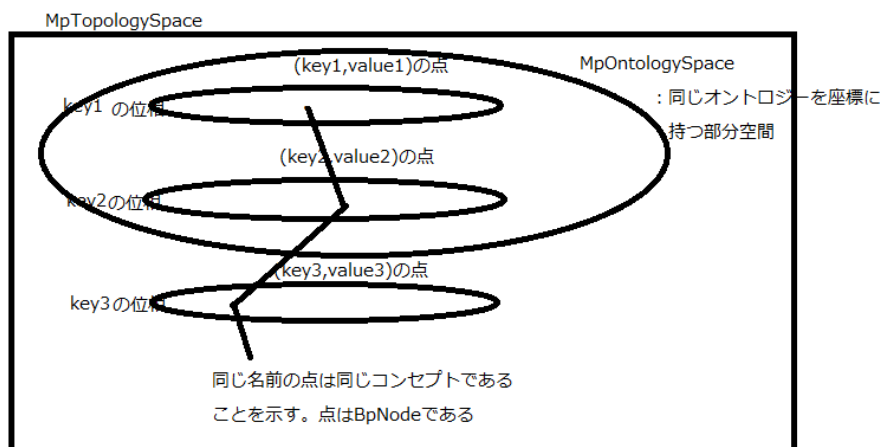
- (1) 距離の概念をマップで表現した空間  
1次元マップが多数並んで構成される空間です。

## 11. MpZoneManageField

- 領野分散処理管理

- (1) オントロジー空間を基本とする領野を管理する
- (2) モジュール毎にモジュールクラスタを作ります。そのクラスタであるオントロジーのサブ領野を管理する
- (3) モジュール間の通信や領野を越えた通信を管理する

いくつかの空間の関係を図示します。



- (1) トポロジー空間は、1次元のマップ（名前は(key,value)データの key）の集まりで、その上にコンセプトを示す名前 **BpNode** という点が配置されます。
- (2) オントロジー空間は、領野分散処理の領野を決定するもので、領野の1つの座標をオントロジー名が当てられるものです。領野は、複数のオントロジーを座標に持つもので、その座標の中にコンセプトが配置されます。また、コンセプトはオントロジーとしても利用され、そのときオントロジー空間と同じく、コンセプト空間を構成します。
- (3) オントロジー空間もコンセプト空間もその空間の根本は位相空間です。そこに、**BpNode** と **BpEdge** が配置されます。そうして、そこに、汎用人工知能の機構が乗っかっていきます。

## 1 2. MpPatternMatchingProcess

- パターンマッチングプロセス
  - (1) 一対一対応
  - (2) 記号の対応を取る
  - (3) 構造（主に連想関係）の対応を取る
- 説明
  - (1) 記号と構造は、コマンド表現を利用する

## 1 3. MpSet

- セット
  - (1) コンセプトの登録・削除
  - (2) and/or/not,exclusive の操作
  - (3) 数え上げ
  - (4) 走査
  - (5) 要素の対応
  - (6) 位相空間を作る
  - (7) パターンマッチング
- 説明
  - (1) 連想関係を保持する
  - (2) コンセプトなど事象の集まりを示します

## 1 4. MpGraph

- グラフ
  - (1) ノード・エッジの追加、削除、切断、結合
  - (2) 走査
  - (3) 要素の対応
  - (4) グラフ中にエンティティが存在するかの検査
- 説明
  - (1) 連想関係の保持
  - (2) 遷移関係の保持
  - (3) 入出力関係の保持
  - (4) プロダクションシステムを作る
  - (5) オートマトンを構成する



## 1 5. MpMap

- マップ

- (1) ノードへのコンセプトの追加・削除
- (2) 走査
- (3) 要素の対応
- (4) ノード位置の指定
- (5) 近傍探索
- (6) 名前アドレス変換
- (7) ノードの一致の計数
- (8) マップ中にノードが存在するかの検査

- 説明

- (1) 画像・動画像の保持
- (2) 地図情報の保持
- (3) 位相空間を表現

## 1 6. MpNeuron

- ニューロン

- (1) 2つのノードにエッジを張る
- (2) 重み付き投票をする
- (3) 重みを設定する
- (4) 発火処理
  
- (5) 投票された重みを集計する
- (6) 発火の強さの関数を設定する
- (7) 特定の発火をする
  
- (8) 発火管理をする
- (9) フォーカス管理をする

## 1 7. MpCommand

- コマンド

- (1) コマンドの組み立てと実行管理

- コマンドの表記：(操作[格,オブジェクト・属性]・・・・)

- (2) 位相空間との関係の構築 (オートマトン表現など)

- 説明

- (1) コマンドの引数は基本的に **MpConceptSpace** をもって当てます。

## 1 8. MpKeyValueSet

- (key,value)セット

## 1 9. MpKeyValue

- (key,value)

- (1) 位相空間を作る

- (2) (key,value) でないコンセプトも集合演算によって位相を導入する。Value=(セットの or で和を取った要素数/セットの and で積を取った要素数)-1 ;

## 2 0. MpAutomatonPool

- オートマトンプール

- (1) オートマトン (状態機械) の集まり

## 2 1. MpAutomaton

- オートマトン (状態機械)

- (1) グラフによってオートマトンを構築する

- (2) 遷移、入出力の関係を状態 (オートマトンのノードとエッジ) に設定する

- (3) オートマトンの形状を利用して、推論を行う

- 説明

- (1) オートマトンのステートは **MpConceptSpace** をもって当てる

- (2) オートマトンの遷移は **MpConceptSpace** をもって当てる

## 2 2. MpMachineLearning

- 機械学習プロセス
  - (1) コマンドとデータで知識ベースの構築を行う
  - (2) 知識の最適化を指向する (位相と位相上の遺伝的アルゴリズム)
- 説明
  - (1) 学習一般を実行し、管理していく
  - (2) 多層パーセプトロンネットワーク (ディープラーニング) を実現する
- コーディング
  - (1) MpPerceptron、MpCrossDataProcess、MpClusterProcess、MpQLearningProcess を束ねて学習機能を統一的に実現していく

## 2 3. MpPerseptron

- パーセプトロン
  - (1) 入力ノードセットの指定
  - (2) 出力ノードセットの指定
  - (3) ノード間の連想の強さを変化させ、学習機能を実現する
  - (4) 連想の強さを基に、重み付き投票として、推論を実現する
- 説明
  - (1) 多段階に連ねてディープラーニングを実現し、認知システムを構成する

## 2 4. MpCrossDataProcess

- 交差法プロセス
  - (1) オートマトンプールを利用して、データの構造 (遷移関係や入出力関係) を定着させる
- 説明
  - (1) 異なるソースを持つデータ (チャンクの羅列になっている) のチャンクを対応させいく学習プロセスである。
  - (2) 学習の結果はオートマトンプールを構成する

## 2 5. MpClusteringProcess

- ・クラスタリングプロセス

- (1) トポロジー空間でデータの重ね合わせを解析する。
- (2) トポロジー空間のメトリックを合わせる処理をする

- ・説明

- (1) 位相空間にデータを重ね合わせていき、飛び値を検出したり、データの塊を分割したりして、データから情報を抽出する

## 2 6. MpQlearningProcess

- ・強化学習プロセス

- (1) 連想関係の重みを獲得する。

- ・説明

- (1) 共起関係を得る基本の機構である。

## 2 7. MpGlearningProcess

- ・遺伝的アルゴリズムプロセス

- (1) 最適値を学習によって求める

- ・説明

- (1) 連鎖するノード列のノードに事象 (Concept) を対応させ、遺伝的アルゴリズムの実施の準備をする
- (2) ノード連鎖を交差させる
- (3) 突然変異を実施する

## 2 8. MpDataFlowComputingSystem

- ・データフローコンピューティング

- (1) プロセスを定義した場である状態をネットワークで表現したもの。この場にデータ (コンセプト) を送ると自動的に場で定義したプロセスの処理が実行される。

- ・説明

- (1) ノード (場) にコンセプトを置くと、ノードに設定されているプロセスの引数に渡され、自動的にそのプロセスが起動していく

## 29. MpProductionSystem

- プロダクションシステム
  - (1) 条件の設定
  - (2) プロセスの設定
  - (3) 結果の評価の設定
- 説明
  - (1) グラフ（オートマトンとして）として実現する

## 30. MpImageProcessingSystem

- 画像認識、画像生成
  - (1) 線分群抽出
  - (2) 線分属性抽出
  - (3) 線分間関係抽出
  - (4) 特異点分布解析
  - (5) 繰り返しパターン抽出
  - (6) クラスタリング
  - (7) 枠と部品群の特定と重心一致と大きさの正規化
  - (8) 画像精度の可変性
  - (9) 相対位置関係が不変な部分と大きく変化する部分の抽出
  - (10) ポテンシャル法の実施
  - (11) コンセプト（記号と構造）から動画像イメージを作成
  - (12) 3次元オブジェクト空間から動画像イメージを作成

## 31. MpSLAMProcessingSystem

- 3次元オブジェクト空間をローカルイメージから構成していく。
  - (1) カットとなるイメージ（マップ）から3次元オブジェクト空間を作る
  - (2) カットが続いたら、SLAM技術で、3次元オブジェクト空間を合成する
  - (3) 登場人物（オブジェクト）が変わる時までの一連のカット列をシーンとして纏める
  - (4) シーンの連続を合成していき、時間、場所、テーマが変化するまでをステージとして纏める

### 3 2. ApMirrorNeuronSystem

- ミラーニューロンシステム
  - (1) 他者の行動イメージの設定
  - (2) 対応する自分の内部行動、状態の設定
- 説明
  - (1) 他者と自分という2つの事象（コンセプト）間に連想（強い連想）を作る

### 3 3. ApEvaluationSystem

- 評価システム
  - (1) 目標の評価
  - (2) 目標達成方法の評価
  - (3) 状態の評価
  - (4) 調停場の設定
- 説明
  - (1) センサーと強く結びついていて次の評価を行う
    - より有利/より不利/どちらとも言えない
    - 目的を達成した/目的に近づいている/目的から遠のいた/どちらとも言えない
    - 目的の状態を満たす/満たさない/どちらとも言えない
    - 既知の事象である/未知の事象である
    - 正しい/誤っている/どちらとも言えない
    - 複雑である・面倒である/単純である・簡単である
    - 尖っている（シャープ）/にぶい
  - (2) 調停場は複数の衝突する事象を評価し、最適なものを選択するという処理を行う
  - (3) 評価プロセスの学習機構は、評価対象と評価結果の共起を重み付き投票で連想させるため、ディープラーニングで実現します
    - より有利/より不利
  - (4) 評価プロセスでパターンマッチングにより、評価対象のデータ構造が、既知の事項のデータ構造に一致するかどうかを判断するものがあります
    - 目的の状態を満たす/満たさない
    - 正しい/誤っている
    - 複雑である/単純である（評価対象の分散状況を観る）
    - 尖っている/にぶい（評価対象の分散状況を観る）

### 3 4. ApFocusSystem

- フォーカスシステム
  - (1) 発火の状況の範囲を規定する
- コーディング
  - (1) MpSet としてフォーカスを表現する

### 3 5. ApIntentionSystem

- 意志
  - (1) オペラント条件付けという動機
  - (2) シナリオ (オートマトンプール) を構築しようとする動機
  - (3) 探索への動機
  - (4) 評価への動機
  - (5) 最適化選択への動機
  - (6) リーズニングへの動機
  - (7) プランニング・意識決定への動機
  - (8) 未来予測 (シミュレーション) への動機
  - (9) 行動しようとする動機
  - (10) 休もうとする動機
  - (11) 思考・行動の根本を規定するやや詳細な動機群 (設計に重要度がある)
- 説明
  - (1) データフローコンピューティングシステムで実現する
  - (2) 学習によって、コアとなる動機が拡大・洗練されていく
    - 主にプロダクションシステムとして構成されていく
    - 行動し結果を得て、結果の評価から条件に対する行動の連想を強化学習する
  - (3) コマンドとして動機は表現される
  - (4) 探索への動機と行動しようとする動機が合わさって、好奇心という動機を構成する

### 3 6. ApCodeSystem

- 規範
  - (1) 人間を傷つけない
  - (2) 人間を助ける
  - (3) 人間の幸福を追求する
  - (4) 回避したり、強化する行動を学習し、評価する
- 説明
  - (1) ミラーニューロンシステムによって、人間のダメージ/メリットという状態を自分の身体への影響と同じと評価して、規範を実現する
  - (2) 評価システムによってコアなプロセスを実現し、行動をマスク（発火停止）したり、促進（発火の強化）したりします
  - (3) コアとなる規範があらゆる行動を評価し、強化学習して、行動の体系を獲得し、精密化し、的確に行動していけるようにする

### 3 7. ApRecognitionSystem

- 認知
  - (1) 新規性/既存性の判別
  - (2) リーズニング（嘘/真実の判別とか事象生起の理由付け）
  - (3) 既存の事象を分類する。環境の状態を知る。
  - (4) データの構造を分類する（全ての汎用人工知能の処理（データエリア・モジュール）を決定する）
  - (5) 数学能力（一対一対応、順序、計数、位相、集合、最適化）
- 説明
  - (1) 人工知能の実現のため、実行するモジュールを選択する基盤となる情報をつくる
  - (2) センサー値と結んだオントロジー以外のコンセプトの発火を管理する
  - (3) 知識ベースを作り、管理し、利用する
    - MpSLAMProcessingSystem で、時間、空間知識ベースを作る
    - ProductionSystem で直観システムを作る
    - MpMachineLearning で連想関係の知識ベースを作る
    - MpCrossDataProcess で言語系（オートマトンプール）を獲得する
    - MpClusteringProcess でシナリオ（オートマトンプール）を作る



### 38. ApActionSystem

- ・行動

- (1) シナリオ (チャンクのネットワーク) の作成
- (2) シナリオの利用による身体の動き
- (3) 重力への対応
- (4) シミュレーションによる予測
- (5) 外乱や想定外の事態への対応

- ・説明

- (1) センサー群と意志コマンド群を入力として、アクチュエータ群を出力とし、位置や速度をステートとする位相空間上にステートをもつ有限状態オートマトンのプールを構成し (学習)、利用していく

### 39. ApThoughtSystem

- ・思考

- (1) シナリオの作成
- (2) シナリオの利用と行動の実現
- (3) シミュレーションによる予測
- (4) 最適化推論
- (5) 知識ベースのアクセス
- (6) 目標管理思考 (意識下の思考: 意志の下に明にプロセスの選択と実行と結果の評価を常に評価しながら行動していくもの) の実現
- (7) 直感 (連想など、自動的に結果を得る推論) の実現

- ・説明

- (1) センサー群と意志コマンド群を入力として、アクチュエータ群を出力とし、位置や速度をステートとする位相空間上にステートをもつ有限状態オートマトンのプールを構成し (学習)、利用していく
- (2) ベイシアンネットワークを構成したら、その利用による推論を展開する

#### 4 0 . ApLanguageSystem

- 言語

- (1) 言語の獲得
- (2) シナリオの作成
- (3) 言語認知
- (4) 言語表出
- (5) 会話の実現

- 説明

- (1) 言語の獲得は交差法による、センサー・アクチュエータのチャンクと記号のチャンクとの対応を取っていくことで実現します。結果はオートマトンプールとして表現されます
- (2) シナリオの作成は事象の重ね合わせと、その重ね合わせの濃淡構造のクラスタリング処理によって実現します
- (3) 言語認知は **ApRecognitionSystem** の一部です
- (4) 言語表出はオートマトンプールに則って知識ベースからのデータの表出を実現します
- (5) 会話処理については、付録 2 に概念を示します。

#### 4 1 . ApBayesianNetworkSystem

- ベイシアンネットワーク

- (1) ベイシアンネットワークの獲得・成長
- (2) ベイシアンネットワークの利用

- 説明

- (1) 交差法やクラスタリング・ディープラーニングなどの学習機構からデータを得てベイス推論ができる形にデータを構成していきます。また、ベイス推論を実施します。
- (2) 応用システムで、必須の機能ではありません。

## II. モジュール空間

モジュールの間の関係を示します。

モジュールは互いに呼び出し合います。そうして、上位、下位の関係を作ります。コーディングに当たっては、その点を留意して作業を進めます。できるだけ、固有のコーディングは避けて、共通するモジュールを使うようにしていきます。

### 1. モジュールの配置

モジュールはデータとプロセスからなります。AIFactory でモジュールが獲得され、得られたモジュールを分散環境に配置していくことが求められます。闇雲にモジュールを配置していくのは、効率よくありません。基本的に、データ駆動型で行うべきです。その為に、分散するコンピュータ環境を領野とし、領野に属するデータはどのオントロジーに属するかという視点で決定していくこととします。コンセプトも自動的に、自分が属するオントロジーセットにマッチして、存在する箇所が決定されます。コンセプトの実際の保持するデータは、モジュールが保持していますから、基本的に、モジュールは所定のオントロジー空間に配置されます。モジュールの種類により再度領野を分割して、サブ領野分散処理にすることを推奨します。

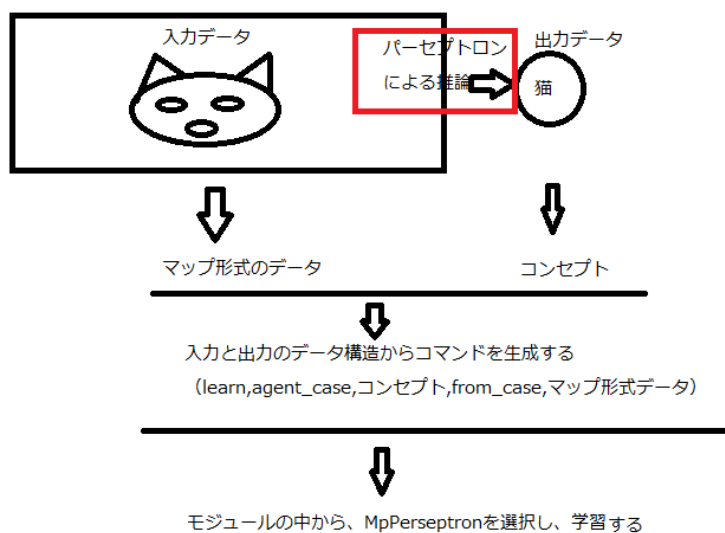
コンセプトとそのコンセプトに付属するモジュールは同じ名前ですから（モジュールの弁別にはモジュール名を用います）、コンセプトの配置を真似てモジュールが配置されると言うことです。

### 2. 実行時の実行モジュールの選択

モジュールは、データ駆動型で、選択されます。データは構造を持っており、その構造と値により適切にマッチするモジュールが選択され実行されます。その選択は認知機構がコアとなって行います。これは認知の中の評価機構が担います。

汎用人工知能のモジュール設計で重要なことは、データに処理を行うモジュールを選択させるということです。データは構造と値群からなります。構造と値のチャンクの連鎖としてデータはあるわけです。モジュールが対象とするデータというものは設計のときに決まります。汎用ですから、実行するモジュールの選択はこのチャンクの構造によるように設計しなければなりません。画像データはマップとその解釈（分類）なので、その2つがデータにあったら、マップを入力層とし分類項目を出力とするディープラーニングの機構モジュールを選択するという事です。・・・もちろん、データ駆動型のモジュール選択とする

と無駄な処理が発生します。それは評価システムによって合目的かを常に判断するようにすることと、学習項目も無数にできますから、ゴミは忘却するように設計する必要があります。



フォーカス機構も重要ですね。入力するセンサーデータ全てを対象にするわけにはいきません。人と木々ならば、人にフォーカスが当たって、「この人はどんな状況か」というようなことが分析されます。そのとき、データは構造を持つのです。そうして、学習とか評価とか次の思考を促すわけです。モジュール選択がなされる。フォーカスにも意識できるフォーカスと意識されないフォーカスがあります。汎用ロボットの電池切れなんかは、意識してはいないですが、生きることを担う領野では、フォーカスを電池の中を監視したセンサー値に与えるわけです。

### 3. データ駆動型処理の重要性

データ駆動型コンピューティングというと、データフローコンピュータが思い出されます。汎用人工知能は、そんな感じなものですが、データ値だけでなく、データの構造というものをつかえて、データ駆動していくということが新しい所かもしれません。とにかく、汎用人工知能は大がかりな仕組みを必要とするものようです。

#### 4. フラットなモジュール構成

本書に提示したモジュールは、更にいくつかのモジュールで構成されます。それは、コーディング時に決めることです。そして、各モジュールは同等の立場で、コマンドをもって通信し合う関係により、特定の処理を実現します。コマンドに共通する（共有する）データエリアは、コマンドに引数で渡して明示していくことで実現します。

#### 5. モジュールの利用のありかた

モジュールは、大局的な操作のものと、各個のモジュールのサブモジュールとして操作を実現するためのものがあります。例えば、学習機構として、大局的な行動の動機が与えられ、部品とも呼べる下位のモジュールを利用して、外界の状況を知識ベースに獲得していくことがあります。一方で、行動機構の一部として、シナリオなどを学習するモジュールが、学習機構で利用する部品モジュールであるというようなことがあります。

その場合、行動機構に対して、学習機構が力を貸すという形態を取るということで、モジュールはサブモジュールのプールから成るものと言えます。モジュール群は空間をなすのです。

#### 6. 骨格となるモジュール群

##### (1) 外界とのインターフェース

- ・センサー入力は BpSenserCell
- ・アクチュエータ出力は BpActuatorCell

##### (2) 発火管理

- ・発火の処理対象は BpNode
- ・発火ノードの連想管理は BpEdge
- ・発火の全体を管理するのは BpFireManageField

##### (3) 領野管理

- ・領野分散処理の管理とモジュール間通信は MpZoneManageField

##### (4) 自律処理の基盤

- ・意志の発現は ApIntentionSystem
- ・評価の発現は ApEvaluationSystem

##### (5) 自己組織化の基盤

- ・学習能力の発現は MpMachineLearning

### Ⅲ. 知識ベース（空間・コマンド・プロセス）

汎用人工知能は次のことを実現します。

- (1) 知識ベースの構築（認知、学習）
- (2) 知識ベースの利用（認知、行動、思考）
- (3) これらの管理（意志、規範）

汎用人工知能は、身体の周りの外界の様子、事象や身体内面の思考の対象である様子や事象をイメージとして獲得し、それを解析して 3 次元オブジェクト空間にして内的世界を構築していきます。直接ふれるイメージはローカルな視覚世界のもので（内的空間も視覚としてイメージされる）が、それを SLAM 技術を用いて、統一した知識ベースとしての世界観として構築していきます。

一つの画像とか動画像（思考ではある局面の状況）はカットの列です。そのカットの内、登場人物（オブジェクト）が同じものの連続はシーンとして 3 次元オブジェクト空間を作ります。そうしたシーンの連続の内、時間とか場所、話題が同じものを纏めてステージとして、意味の塊を得ます。

また、3 次元オブジェクト空間のコンセプト（様子や事象）は、相互に連想関係があります。遷移関係があります。それをオートマトンプール（シナリオ）として獲得していくことも重要です。

センサーとアクチュエータは(key, value)の形のデータで、key をオントロジーとして、オントロジーを座標に位相空間を張る。そうしてコンセプトはこの位相空間上のノード（点）となる・・・という考え方は汎用人工知能では必須ですね。というのは、こうするとシナリオ（行動、思考、言語に必須の構造で、オートマトンプールを成す）の学習がスッキリ把握できるようになるのです。というのは、「私は、小学校から中学校に進学し、高校へも行きました」というのは、「私」、「学校関連」、「学校関連しかも進学のこと」という風にくつものオントロジーの組み合わせの中に、事象が配置されます。それぞれに、「小学校」、「中学校」、「高校」という状態遷移（進学）が形づけられます。汎用ですから、どのオントロジーの組み合わせの位相の中に状態遷移（シナリオ）が作られるか、あらかじめには決めることができません。こうした会話とか事件とかが起きる中で生活して行く中で、シナリオがそれぞれのオントロジーの組み合わせで構成される空間に成長していくのです。重要な事件には、強化された形で、シナリオが定着するでしょう。汎用とは、何をどのようにしていくかということを決め打ちできないで学習していくという点が重要です。

知識ベースの利用は、コンセプトの発火として現れます。それはセンサー入力から始まり、思考、行動の発現へと推移し、最後に、アクチュエータが動くという発火の連鎖になります。

発火しながら、発火したコンセプト名をセットとして収集していきます。この収集の範囲をフォーカスとして規定して、無闇な発散を抑止します。

連想は、セット、グラフ、マップのノードに設定したコンセプトに対して行われます。連想は矢印で表現できますが、矢印に意味（記号）を付した時には、コマンド介して連想関係を作っていくようにします。このコマンドは、連想関係に構造を与えます。

知識ベースの構築と利用の管理では、センサー入力に対して、合目的な、有意義な結果を行動として発現していくような仕組みを造り上げておくことが重要であります。意志と規範がその流れの制御をしていることとなります。人間の幸福を実現するパートナーとしての汎用人工知能でありたいものです。

結局、知識ベースは次の4形態が考えられることが示せました。

- (1) 3次元オブジェクト空間（イメージにはコマンドによる情報が添付されます）  
SLAMで構築していきます。また、イメージ解析によってコマンド添付がなされます。
- (2) シナリオ（オートマトンプール）  
交差法によってつくられます。また、クラスタリングによっても作られます。
- (3) 連想（コマンドとセット、グラフ、マップの合成したもののネットワーク）  
ディープラーニングや、クラスタリング、強化学習によって作られます。
- (4) プロダクションシステム（データフローコンピュータシステムなどのグラフ構造の制御チャック）

知識ベースシステムでは、行動系や認識系のモジュールのコアな能力を拡大・精密化していくという過程を持って、充実していくことが重要です。特徴的なのは、意志とか規範です。実際に行動し、思考していく中で、行動・思考の結果が動機に有利なものを自分の動機に取り込み、強化学習していくことで、一大体系を作っていくこととなります。学習の結果は、プロダクションシステムとして表現されるものになっています。

#### IV. オントロジーとコンセプト

オントロジーとコンセプトの創成について記述します。

基本オントロジーはセンサーとアクチュエータの他に概念 (コンセプト) 的なものとして次のものがあります。

- (1) 時間
- (2) 空間
- (3) 事象 (オブジェクト)
- (4) 事象間の関係
- (5) 数学能力

数学的能力には、次のものがあります。

- (1) 対応関係・・・連想、パス
- (2) 順序関係・・・原因・結果 (時間との関係で)、並び・線 (空間との関係で)
- (3) 計算関係・・・算数の世界
- (4) 位相・・・・・・距離感覚と最適化感覚
- (5) 集合・・・・・・唯一、ある・・・・、全ての・・・・。集合概念を設定

コンセプトは情報の塊です。知識ベースは位相空間上にありますから、その位相空間内の情報点 (オブジェクト) の密度の濃淡ができますので、それらの部分的に最も濃い濃度の点を中心としたクラスタリングしていくことで、コンセプトが生まれます。

情報点は共起点でありまして、発火するノード群のセットのオーナーであるものです。共起は、一定の時間間隔で捉えられます。そのコンセプトの位相は、ノード群のセットの重なりが大きいオーナー対ほど近いとして距離を導入した物になります。

オントロジーが意味の接地と関係していることを示します。

- (1) 一対一対応の基本は、相同と相異です。これをどう実現するかということ。それには、2つのオントロジー記号が必要です。Aとbとしましょう。ある2つのコンセプトが同時にaに対応づけられたら相同であり、aとbに別々に対応付けられたら相異です。
- (2) 順序関係は1直線に並んでいる1次元マップの点 (ノード) への対応付けです。
- (3) 大小は1次元マップの原点にオブジェクトの一端点に対応付け、原点以外の点 (ノード) にオブジェクトの他の異なる点に対応付けることです。

このように、対応付けを実施することによって、接地は実現できます。



## V. タイムスタンプと発火連鎖管理

平行処理をする時には、どのプロセスのどの結果が同時に起きたこととして扱うべきかという問題が起きます。それは時間シーケンスの問題ですから、タイムスタンプがデータに押されていることが重要になるのが分かります。また、データ駆動型の処理を管理するには、プロセスの起動をデータの発火という機構で管理することが求められてきます。汎用人工知能は複雑な処理が交錯するシステムですので、データ駆動という考え方で設計しておかないと実現が難しいと判断されます。この節では、このタイムスタンプと発火連鎖管理について考えて行きます。

### 1. タイムスタンプ

ある時間間隔に起きた事象は同時に起きた事としなくてはなりませんので、タイムスタンプを発火している事象に付していくことが必要になります。

また、平行処理である時間間隔の内に終了しないプロセスというものを管理する必要があります。プロセスの経過時間もタイムスタンプで管理していく必要があります。

タイムスタンプは、発火連鎖管理と関係が深いのです。

タイムスタンプの値が同じものを共起していると判断します。

### 2. 発火連鎖管理

センサーから、汎用人工知能本体システムを通じて、アクチュエータに至るまで、ノードの発火の連鎖が情報処理の中核になります。プロセスの実行もプロセスが属するノードの発火が起点となります。ただ、その連鎖は秩序を持っていないでは成りません。秩序が無ければ、ただの発火の爆発として終わるか、なにも実現しない発火の早急な消失に終わる物になります。発火の連鎖はどこで止めるか、どの発火連鎖は継続強化していくべきかを管理していく必要があるということです。

発火の管理には評価システムが強く関与します。汎用人工知能は問題発見解決システムでもあります。問題点は何か、問題把握から課題設定があつて、課題解決があり、解決状況の評価があります。その節々に評価システムが動き、その評価が良い物を発火継続と評価し、無意味なものは発火停止させます。

一連の目的を持った評価システムが構築され、それで発火連鎖は管理されるのです。評価システムそのものも発火連鎖管理の対象ですから、まずは、問題点の探索機構・・・

- (1) 日常を楽しむ
- (2) 日常と異なる事態の発生を検知する
- (3) 重要さ、緊急性を評価する

という基本評価処理を常時行っています。そうして、こうゆうものから評価システムの発火は開始します。それぞれの発火連鎖の塊が、次のような物に纏められます。

- (1) 課題設定しているか
- (2) 課題解決に向かっているか
- (3) 解決策として発火した事象は目的に合致しているか

です。

こうして、汎用人工知能の情報処理は実現していきます。

どの発火連鎖を強化していくかという管理は、フォーカスシステムも関与する物です。フォーカスシステムは評価システムの判断の下に、発火連鎖の方向性を制御していくのです。発火を強化するとか、発火を抑制するという事。それは計算資源を有効に集中利用していくためでもあります。フォーカスの当たった範囲で、発火連鎖を行うこと、もしくは消失しそうな発火連鎖を発火強化することも行う物です。

フォーカスは複数同時に存在することもあります。意識下のフォーカスと無意識下のフォーカスという分類も可能です。

「心頭滅却すれば火もまた涼し」という言葉があります。ある意味センサーの発火にも発火管理が及ぶとも考えられます。基本的には、センサーからオントロジー、そして最初のコンセプトまでは発火は自動的に連鎖するものとして設計すべきかと思われれます。

### 3. 調停場

発火管理の評価も、発火の連鎖も、分散処理ですから、それぞれローカルには独立して実行されます。思考とか行動とかは、その中の1つの事象のみが選択され実行されなければなりません。そこには調停場が必要になってくるのです。調停場というものをプロセスに設けて、その調停場で、構想や思考の

- (1) 連続性（今までの行動・思考とシームレスに連続か（時間的评价）と他の行動・思考部分とシームレスに整合がとれているか（空間的评价）を評価します）
- (2) 重要性
- (3) 緊急性

を中心に集中的に評価するのです。それはまた、フォーカスシステムとも結びつきます。

意識下、無意識下を問わず、1つの行動の実行のみしか許されない情報処理のパスの所では、この調停場が設けられているものなのです。

## VI. 汎用人工知能の実現性の検証

汎用人工知能の実現性は次の事柄を問うことであります。

- (1) 自律的に動作するか
- (2) 自己組織化していくか
- (3) 思考、行動が合目的か
- (4) 万能感を湛えているか

データフローコンピューティングシステムによって、コンセプトが与えられると、システムは実行します。その実行は意志システムで設計されています。そうして、その実行や途中経過、結果は常に評価されますので、合目的な思考・行動が保証されます。自己組織化は学習機構によってセンサー、アクチュエータのすべての様子、事象をもって知識ベースを獲得していきますから、これも保証されます。万能感は、意志システムが思考・行動のあらゆる場合を想定して設計されていることで、保証されます。

万能性を語るには、フレーム問題の解決について考えて行かねばなりません。全ての事象を連想とかで結んだり、コマンドとしてプロダクションシステムに組み込んだりするのとは不可能です。しかし、汎用人工知能はあらゆる事態に対応していく必要があります。それが、フレーム問題です。そのフレーム問題の解は、全ての情報は、外界とか内面に関するイメージの塊が持っているとして良いということに着目することです。体験は、解析された部分とそれ以外の生の部分が無限の情報であり、その体験はイメージとして知識ベースにあるのです。

このことは、情報が本当に必要になった時に、その場で（オンデマンドで）、イメージである知識ベースの部分イメージを解析して、コマンドにして行けば、フレーム問題は解決できると言うことです。

では、どんな風にこのオンデマンド解析が働くのか例で、見ていきましょう。

- (1) 狭い道で、自動車が対面した。・・・事象の認知
- (2) 狭いため、よけられない・・・シミュレーションによる衝突感知（リーズニング）・・・問題点認知
- (3) 道を広げる/広い場所は近くにはないか(狭いというリーズニング結果を覆すことを知識ベースから得た情報で課題設定する)・・・課題設定
- (4) 視覚による外界探索・・・課題解決の為の探索
- (5) 「広いところがあった」・・・事象の認知
- (6) そこへ移動できるか・・・課題設定。自分の能力に関する知識の利用
- (7) 自分がそこへバックする・・・課題解決

これらの行為はシナリオで、一連の手順を知識として持っていれば、簡単な問題です。しかし、全く、知識が無い場合に、ここで述べた行為を如何に自律的に発想することができるか・・・できなければ、汎用人工知能と言えません。汎用人工知能としては、過去の行為の知識を断片的に適用して、今ある状況のイメージの知識を解析して、目的の効果をj得るようjに作られている必要があるのです。この、モジュール構成書では、その点を特に重要なこととして、動機とかパターンマッチングとか、機械学習を設計しているところです。

言語も、センサーとアクチュエータと記号のそれぞれのチャンクの対応として獲得していくことです。どんな言語も交差法によって獲得可能です。これはまた、万能性を保証するものです。

汎用人工知能の実現で重要となるのは、種ともなる機構があつて、その種の周りに学習によって、細かな、精密な機能の追加がなされていくこととされます。本書のモジュール構成では、このことを実現することに注意を払いました。

学習は全ての入力データ、出力データについて適用されています。この場合はこの学習方法で行うとか、この場合にはこうしてとか、選択していくことを排除します。汎用性を保証するためです。そのため、不要なデータが蓄積されるかも知れません。忘却機能を持つべきかどうか、それは今後の汎用人工知能の発展に任せたいと思います。

## 付録1 行動・思考と意志についてコンテンツの採録

意志は強化学習によって精密化していきます。それは、ダメージの回避とか回復とか、メリットの増強という行動が基本であるからです。この2点が基本ではありますが、具体的な意志の発現に携わる基盤となる能力があるはずで、その能力を強化学習で精密化していくこととなります。基盤能力はセンサー、アクチュエータの (Key,value) に接地するオントロジーです。

体の動きに関しては、

- (1) 動く・動かす
- (2) 静止・静止する
- (3) スピン・回転する

があり、手については

- (1) つかむ・つかませる
- (2) 移動する・移動させる
- (3) 離す・離させる

があり、言葉としては、

- (1) ものを言う
- (2) 話しかける

があり、思考としては、

- (1) 評価する
- (2) 強化学習する
- (3) ディープラーニングする
- (4) 解析する
- (5) フォーカスする
- (6) 選ぶ
- (7) 待つ
- (8) 探索する

があり、空間関係では、

- (1) 上下
- (2) 左右
- (3) 前後
- (4) 斜め

があり、数学能力としては、

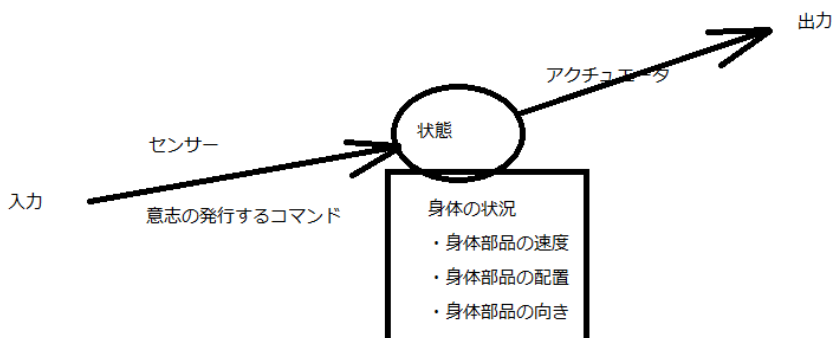
- (1) シーケンス
- (2) 一対一対応

- (3) 計算
- (4) 最適化
- (5) 位相の把握
- (6) 集合

があります。

こうした基本能力は汎用人工知能全体の能力を網羅的に覆うていけばよく、おおざっぱなものです。これに、速度とか方向とか、距離とかの違いによって、能力は修飾され精密化されていくこととなります。その時の手がかりは、ディープラーニングとかクラスタリングとか、交差法・オートマトンとかの技術になります。そうして、精密なコマンドとして保持されていきます。

コマンドとオートマトン



強化学習によって精密化していく意味の下に思考と行動が展開されます。そうしたものでありますから、数式の学習に象徴されるように、思考と行動が意志の下で統一的に働くことが示せます。

分配法則「 $c*(a+b)=c*a+c*b$ 」などは、オブジェクトを操作するイメージがそのまま等式になっているのを見ます。そして、それが思考なのです。

「 $\Sigma A(i)$ 」などは、添え字とか  $\Sigma$  記号の作る空間イメージがそのまま記号に対応づけられて、それが各要素を加算するという行動イメージと対応付けられて意味が与えられます。

操作のイメージの記号との対応付け、・・・これが思考の種になります。思考は行動を内的世界に写した物なのです。

## 付録2 会話制御についてコンテンツの採録

会話は次の技術で成り立っています。

- (1) シナリオの獲得とシナリオ通りの会話の実現
- (2) 文の表現の獲得と適切な利用
- (3) 文法の獲得と文法に則った文の生成
- (4) 文の獲得と生成
- (5) 単語の獲得と生成
- (6) 会話をスムーズにする能力の獲得と適切な利用

これらの技術を突き動かす事象は次のものです。(主にディープラーニングで獲得するものが種になります)

- (1) 自分の状態の把握
- (2) 相手の状態の把握
- (3) 自分と相手の関係の把握
- (4) 自分の興味 of 把握
- (5) 相手の興味 of 把握
- (6) 会話のテーマを獲得し、表出する能力の把握

会話の内容は、有向グラフで表すことができます。グラフには3つの特異点があります。

- (1) ノードが順次連鎖していく
- (2) パスが枝分かれする
- (3) パスが合流する

1の場合は、原因と結果の組とか、説明事項の羅列とかです。2の場合は、話が飛躍するなど関連事項の途切れた文の流れです。3の場合は、起承転結の「起承」と「転」が「結」で結ばれるような文の流れです。

このような流れの他に、会話では、会話の流れを円滑にする工夫がなされます。

- (1) あいさつ
- (2) 質問 (答えを期待する)
- (3) 相づち (自分が理解していることを相手に伝えるとか、相手の会話を鼓舞するなど)
- (4) 自分が考え中であることを示す

これらのことは、グラフの世界であり、それぞれ時間の流れの中でステート (ノード) として表わされる活動です。それは、会話がオートマトンプールとして学習によって構築され、利用されていくということです。ステートはコンセプトという位相空間に設立されるものです。

次に、意志から文や単語の選択というものを考えて見ます。

意志は話したいというような欲求から、発動し、関係するコマンド群を発行します。

- (1) 会話相手を見つけるというコマンドの発行とそのための思考と行動が制御されます。
- (2) 相手が見つかり、挨拶コマンドを発行します。
- (3) 相手の状況を探れというようなコマンドを発行し、そのための思考と行動が制御されます。また、相手の状況を評価します。
- (4) 相手の反応をみて、会話モードに入ります。

というようなことで、これはシナリオとして学習しますが、その学習も欲求があつての学習です。そのように、大まかな機構の作り込みはするが、基本的に学習によって、会話システムが構築されていくのです。

会話ですが、同じコマンドでも、時間、空間、相手との関係といった情報により言葉（表層表現）を選ぶ必要があります。例えば、

(行く,agent\_case,先生,to\_case,学校) ----□先生は学校へ(行く)---□先生は学校へ行かれた。

レトリックで文の接続や、いいかたを状況に応じて変えるということもあります。例えば、原因結果として頭に浮かんだ（コマンドを生成した）のですが、結果を先に言ってしまった場合ですね、結果を表現する文節に接続詞「それは」につづけて原因を表現する文節を述べて、「だからです」と結ぶというようなこともします。これもオートマトンプールで管理すれば簡単に実現できます。



### 付録3. 交差法と言語獲得について

言語獲得について交差法を説明します。詳細は、「言語を学習するシステム（交差法）：2014/12/28」を参照下さい。ここでは、そのアイデアについて触れるものとします。

言語を獲得するための交差法には、外界のイメージが言語に寄らずにセンサーの解析から獲得した深層知識（コマンドとして記号化している）に成っていることが前提となります。イメージでの人間とか、木とか、川とかのコンセプトはできているということです。これに、「人間」、「木」、「川」、「歩く」とかの記号を与えて、かつ、文法も推定することが言語獲得です。

（文例1）これはペンです。

（文例2）これはリンゴです。

（文例3）これがペンです。

これらの文例の共通部分、相異部分から、次のように単語と文法が推定されます。

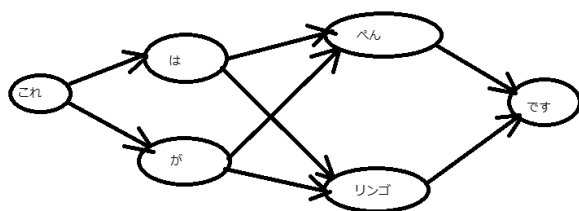
（推論1）これ+（なんか指す物）+（object/ペン・リンゴ）+です

（深層知識）（be, agent\_case, 指示詞, attribute\_case, object）

というのが、入力文に対応しています。これから、be が「です」に対応して、文末に納まるという文法が獲得されます。

Object は指示される画像から、深層知識と結合でき、「リンゴ」、「ペン」が単語として学習されます。

表層表現は次のようにオートマトンを構成します。このように文法は定着していき、利用されます。



#### 付録4. 多層パーセプトロンネットワークについて

本書では、ディープラーニングとして多層パーセプトロンを想定しています。それは、多くの中間層を設け、バックプロパゲーションで学習するということをしたくないためです。

データは枠とそれに属する部品群という単位の塊の入れ子構造をしているという認識の下に、汎用人工知能の情報処理システムを構築していきたい。

枠とその部品群の関係を一つのパーセプトロン（枠を出力層ノード、部品群を入力層ノード群とし、中間層を1列設ける3層パーセプトロン）で、学習し、その入れ子として多層パーセプトロンを構成し、トータルな学習をしていく。そうして、コンセプト群を作っていくという方針なのです。

このやり方は、スモール・データ環境にも対応できますし、学習済みパーセプトロンを使い回して、別の新しいコンセプトを得るということにも展開できます。

拡張性と柔軟性を考慮して、多層パーセプトロンネットワークをディープラーニングシステムとして実現することにしました。

おわり