

1. はじめに

人間は、常に何らかの評価（複数）を下しながら生きています。評価の連鎖は目的を持って整理されたプロセスとして実現し、突飛な行動の発現を無くします。評価があり、思考があり、行動があります。

外界の状況に応じて、筋の通った行動を生成していく過程は、評価システムが整然と動作することによって保障されます。

本論の目的は、この評価システムをいかに構成し、実現していくかを論じることです。

2. 評価の基本事象

汎用人工知能は、自由な環境の中で、生活していきます。センサーとアクチュエータにより環境と相互作用することが生きていくということです。より良く生きる、危険を回避する、何か行動の目標を定め、達成状況の評価し、達成する。こうしたことを実現するために基本的にセンサーやアクチュエータ以外に、論理的なセンサーを設け、評価を適切にくださるようにする必要があります。

以下に、候補の論理センサーを提示します。

- (1) 相同/相異/ニュートラル
- (2) メリット（快適）/デメリット（破壊）/ニュートラル
- (3) 目標事象である/目標成就事象である
- (4) 真実（物理事象）/虚実（論理事象）/ニュートラル
- (5) 自分のこと/他者のこと（相手、第3者、環境）/ニュートラル
- (6) 単数/複数/空
- (7) 時間（過去/現在/未来）
- (8) 空間（1次元/2次元/3次元）
- (9) オブジェクト/運動/属性
- (10) 重要度（各評価事象の強さを認識する項目）

その他、生活する中で獲得した事象をもって評価項目として立てることもします。例えば、ドイツ語の性などは典型です。「性」の接地は、人間の世界とか動物の世界で、子どもを産むというようなイメージとか、性格のイメージとかのイメージのセットとして、連想できるようになって、行われます。

相同/相異の評価の仕方を考えてみます。相同と評価したとき発火するノードと、相異の時発火するノードがあって、パターンマッチングの結果はこの2つのノードのいずれかを発火することで完了します。

その他の評価項目も対応するノードが設置され、評価の結果、適当なノードを発火させることで評価のプロセスは完了します。

評価システムの学習による能力向上を考えてみます。

「りんごを食べる」という事象は、自分の空腹であるというセンサーの状況変化から「メリット」という評価事項を発火し、「メリット」と「りんごを食べる」、「りんご」、「食べる」を連想関係で結びます。

次に、「みかんを食べる」では、「メリット」と「みかんを食べる」、「みかん」、「食べる」を連想関係で結びます。

ここで交差法から、「(メリットのあるもの)を食べる」は「メリット」であると抽象化が進みます。こうして、パターンマッチングの条件部分が抽象化されていくという学習が進みます。

次に、「毒キノコを食べる」では、体を壊しますから、「デメリット」が発火します。「毒キノコを食べる」、「毒キノコ」、「食べる」は「デメリット」に連想されます。これを抽象化して、「(デメリットのもの)を食べる」は「デメリット」となります。

こうして、評価システムは、外界に対して柔軟な対応ができる、より強力なシステムへと能力の向上が図られます。この一連の抽象化のプロセスは交差法と呼びます。

事象（オブジェクト）は属性のセットで表現されます。オブジェクトの属性は、可能性の強さが基本的な値になります。この可能性の値をマップに持って、汎用人工知能の情報処理は行われます。つまりデータはマップで表現されるということです。一方、プロセスは、小さな処理単位（チャンク）の時系列として実現します。チャンクには、最小単位のものがありますが、いくつかのチャンクが集まって、疑似チャンクを為すという入れ子構造をしています。チャンクの切れ目で、評価システムが、チャンクの処理結果をマップで得て、動きます。

このマップは1つではなくて、複数の可能性のマップがセットとなって、アンサンブルとして（ポピュレーション符号化して）実際の事象を指すこととなります。

3. デジタルツイン（イメージ）とコマンド（イメージの記号化）

デジタルツインは、論理事象を点（時には面）、論理事象の間の関係を線で表現されます。そうして、センサーとアクチュエータによる接点のデータは、ビット列とかビット配列とかのデジタルイメージで表現されるものです。

また、空間的に配置された事象群は、空間を点として、これに付属するビット配列（マップ）上に配置されます。

デジタルツインはグラフとなるイメージですが、これは記号によっても表現され汎用人工知能では基本的なものとなります。点は動詞と名詞・形容詞・副詞で記号化され、線は格として記号化されます。そうして、コマンドを形成します。コマンドは次の形式となります。

（動詞[格,名詞/形容詞/副詞]・・・・・・）

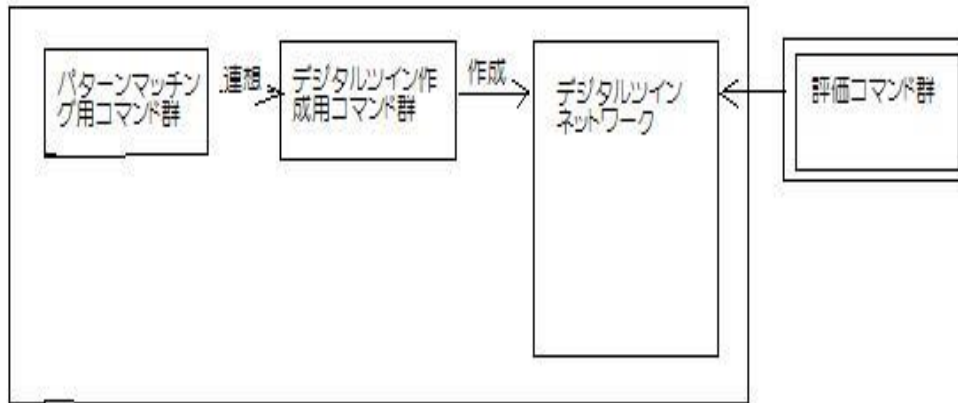
プロダクションシステムで、デジタルツインを構成していきます。プロダクションシステムの中の1つのコマンドは、デジタルツインの全体の中の1つのノードに接続する複数のエッジとエッジに繋がるノードから構成されるグラフを作成していきます。

プロダクションシステムは、パターンマッチングがなると実行するコマンドを定義していくという、コマンド実行のセット・・・・一つの固まりを構成します。パターンマッチング部分もコマンド表記が利用されます。

プロダクションシステムで構成されていくデジタルツインは、時間的に展開していくのと、空間的に広がっていくという2つの方向性があります。時間的に変化していくものとか、相反する事象を表現するノードは、今までの部分デジタルツインが作る空間とは時系列の展開で、別の部分デジタルツインとして構成していきます。複数のコマンドによるデジタルツインでも同じシーンのものは同じ部分デジタルツインとして構成していきます。これは、言語系の文章の構成と同じ処理になります。カットとシーンとステージという構造を持った文章の表現で構成される空間と同じということです。

カットは単一の行動のイメージで、シーンは登場者セットによって決まるイメージの区切りで、ステージは場所（空間）と時期（時間）のある範囲に起きるまとまった事象群の区切りを示します。プロダクションシステムはそうした纏まりのある事象セット毎に作られていく制御システムです。

プロダクションシステム



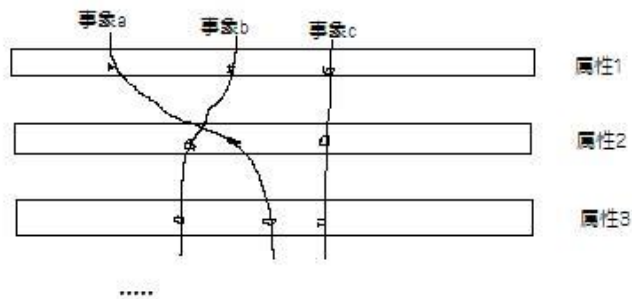
4. SOM 型位相空間とマッチングプロセス

センサーやアクチュエータはもとより、記号も位相を持たせることができます。それぞれの属性を抽出することで行います。記号はもともとのセンサーやアクチュエータ値の属性で1次元の SOM（自己組織化マップ）型位相を埋め込みます。

属性としては、画像の場合、曲率とか線幅、線長とかが考えられます。この点は画像処理の議論に任せたいと思います。

1次元の位相が与えられれば、その属性について近傍探索すると最も近いオブジェクトが選出されます。これがマッチングになります。このような位相は沢山ありますから、高速にマッチングする手法を編み出さねばなりません。例えば、何らかの予測であるオブジェクトが想定されるならば、マッチングは想定されるオブジェクトの属性群の近傍を評価し、一致度をえることになります。

位相空間の構造



事象のマップ表現も、この SOM により実現します。

事象として、例えば、「歩く」、「走る」、「掴む」、「もたれかかる」といった場合を取り上げて、それに位相を埋め込むことを考えてみます。「歩く」と「走る」は、「歩く」と「掴む」よりも近いですね。それは、足の動作と手の動作というように体を基本とした位相になります。属性もそのようにデジタルツインから想定していくことも重要になります。「もたれかかる」は身体の特定の部分である胴体という属性で位相が埋め込まれます。

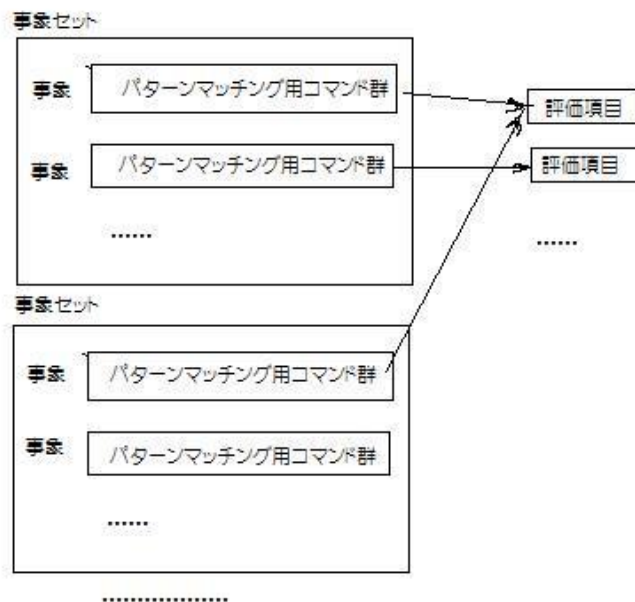
5. 評価システムの働き

評価の結果は、評価すべき直接的な事象の形の他に、その事象が起きている場の状況に深く関係しています。例えば、「真実」ということは、メリット/デメリットの立場からはニュートラルですが、「宝くじに当たったのは真実です」という文の下では、メリットとして強く評価します。しかし、「昔の話ですが、宝くじに当たったことは真実です」では、メリットの強さは、そんなに高くなりません。場の状況に応じて評価は変わってくるのです。

この評価の制御を行うために、デジタルツインという空間を考えることに意味が出てきます。「昔の話ですが、宝くじに当たったことは真実です」は、「昔」という、ずっと遠い過去のことを示す場というものの中に、「宝くじが当たった」という事象が点として存在するというデジタルツインのイメージです。この「遠い過去」の中での真実ということのメリットの評価は、古びていて落ちるという感覚と連想されるのです。そうして、「真実」ということが、この場についていますので、メリットであることは保証されます。こうして、評価は実現していきます。

評価システムは階層構造をしています。例えば、「昔」、「今」、「未来」といった事象セットの中に、「宝くじが当たる」という具体的な事象が置かれます。

評価システムの構造



パターンマッチングをする評価プロセスを事象の発火と捉えると、事象セットの選択は、そのセット内の事象の発火を起きやすくする（バイアスをかける）ことに当たります。

6. 予測シミュレーションと思考の働きと言語化

デジタルツインの下でシミュレーションで予測したり、思考をしたり、言語を発したりすることを行います。デジタルツインは時系列のイメージですが、一連の連続関係にあるものではなく、ステージ、シーン、カットといった断片として切れ切れになって存在するものです。そうして、条件となるデジタルツインの状況パターンとその条件に連想する一連の処理時系列の流れを切り出して、束にまとめたものがプロダクションシステムで、これで、予測シミュレーションや思考（特に直観のところ）や言語化が行われます。

言語も単語をグラフのノードとして、デジタルツインとして生成していきます。言語は、ある条件では文とか節とか単語とかが活用（変形）します。並び順も活用（変形）として変化します。その活用のプロセスもプロダクションシステムとして実現していきます。

言語生成でのプロダクションシステムは、文（単語列）生成用と文や単語の変形用のものがあります。まず、単文として適切な単語列を生成します。その後、単語間の関係とか、文の間の関係とか、デジタルツインの情報を条件として、単語と単文を変形します。

シミュレーションの例として、自動運転を考えると、カーブで車が走って来るのを見たとします。初速は観測した走りです。

そのままカーブを巡る車の運動をチャンクの列で予測します。自分の所に近づいてくる予測イメージの所では、相手の車の大きさ（幅）と自分の車の幅がすれ違うだけのゆとりがあるか評価して、すれ違えるかと判定します。判定の結果、すれ違えると推測すると、そのまま運転を続行しますが、すれ違えないと判断すると、それに対応する運転プロセスを起動します。

シミュレーションは、現在の状況から、未来の最適な状況を予測し（チャンクを逐次選択しながら最適性を評価します）、最終評価から最適な一連の流れを得る・・・予測がうまくいく手続きを発見するというモンテカルロ法の実現でもあります。

思考もシミュレーションの結果を評価しながら実現するのが普通の過程です。現在の状況を解に向けて変化する方向に思考していくから、予測が必要なのです。

言語化もデジタルツインの一部を言語で表現したい状況を、一つのストーリーにすべく、先立つ状況をトレースしていく過程が必要になりますから、シミュレーションは必須なのです。

7. セットとしてのプロセス・データの管理と実行処理

評価は、無数の入れ子構造、シーケンス構造、同時発生構造を持った事象に対して、いくつもの評価項目を立てて実現するものですから、超平行処理をしなくてはなりません。シミュレーションも思考（特に直観は）も言語化も超平行処理の結果を纏めていく過程が必要です。特に、パターンマッチングの部分は超平行処理にし易いところです。

この平行処理をいかに管理していくか。また、学習によってこの能力の向上を図ることも必要ですから、平行処理の管理という問題は重要な技術となります。

その技術として、プロセスとデータのそれぞれをセットとして管理し、さらにそのセットの組み合わせとして管理していくというのが提唱されます。

平行処理に関するセットは、プロセスへのパラメータのセット、プロセスであるプログラムコードのセット、参照するデータのセットです。パラメータのセットは、プロダクションシステムのパターンマッチングする条件に接続するものです。プロセスセットは、パラメータは異なるが計算素子（CPU）にローディングするプログラムは同一であるもののセットとしたり、パラメータ共に異なるプロセスのセットにしたり、パラメータセットを指定されていて、その中で同時に動くプロセスのセットにしたりします。プロセス間の通信は無しで良いはずです。また、プロセスの wait 処理もありません。デジタルツイン空間がプロセスの待ち合わせ場所になるからです。こうして、データは階層を為したり、グループを為したりしますからセットとして管理することが自然になります。

汎用人工知能は、意志とか、思考とか、行動とか、言語処理とかの異なるデータセットを扱う部分に分かれます。これを領野分割と呼びましょう。異なる領野のデータを指定するのは、物理的なデータ存在場所（アドレス）で行うことはできません。記号化（データの名前）で参照しなくては成りません。領野毎に、名前・アドレス変換テーブルを持つこととなります。

また、各領野の中で、現在どういうプロセスをどういうデータで動かすかということを経動的に管理していくことが重要になります。超平行処理を如何に実現するのかという問題です。プログラムとデジタルツインの参照すべきものとパラメータ群を如何に CPU と結びつけるかということです。VLAN スイッチによってバスを動的に管理できるようにすることも必要になってくるでしょう。CPU セットにプロセスやデータを割り当てることは、CPU の実行（事象の発火）に向けたバイアス向上ということになります。バイアスを掛けるコマンドと発火を指示するコマンドの体系があるのです。

なお、「歩く」という事象（コンセプト）は、思考領野にも、行動領野にもあります。また、その「歩く」の下の詳細事象群は思考領野と行動領野で異なり、それぞれのイメージを構成します。

8. 思考と行動の実現への評価システムの展開

意志の下、思考が、そして思考の下に行動が行われます。意志は外界からのセンサー情報からセンサーに連想する行動起動用のコマンドが発行されることで現れます。意志は行動選択のための評価システムにより、適当な行動になるように制御していきます。このあたりのコマンドの動きは思考として現れます。

例えば、飛んでくるボールをラケットで打つ動作を考えてみます。

体育の時間で、ボール打ちをやろうということ、意志は体育の運動とその中の「ボール打ちをやろう」という行為（意志の発生）を選択します。ラケットを持つなどのコマンドとボールを打つという構えを設定するというコマンドへと展開していきます。そのとき、行為の継続性があるかという評価がなされています。

ボールが投げられ自分に近づいてくると、ボールの位置と速度から、ボールの軌跡をシミュレーションして予測します。そうして適切な時点で、適切な腕のスイングを開始します。ボールの予測位置と腕とラケットの動きを常に予測しながら腕の筋肉と手の筋肉を制御していく細かなコマンドを発行していきます。

ここでの一連のコマンドは、デジタルツイン上での学習によって獲得していくことになります。学習には評価システム（失敗、成功などの評価項目で）が動きます。

学習は、事象の重ね合わせの濃淡によって、濃いところ、薄いところと特徴を切り出すところから行われます。そうして、その事象の共起関係から、連想関係を導きだしたり、コマンドの形式による記号化を経て交差法を用いて抽象化概念を学習していきます。

思考も行動も言語の生成と同じように、生成部と変形部の組み合わせであると考えられます。

思考では、今ある課題を解決する行為を実施し結果を作業域に持つ（生成）とする部分と、その作業域のなかの結果事項を、全体の状況から評価し、更なる検討を続ける（変形）部分から成って、目標の問題解決を図っていく過程であります。

行動では、身体の状況と身体の周りの環境と意志（思考を先導する）の状況から、大体の手、足、胴の動きを生成します（運動野とか補足運動野など）。その結果を外界のセンサー状況で微調整する過程（変形）から成ります（小脳系）。

このように生成部と変形部という仕組みを導入すると、システムがスッキリしたものになります。

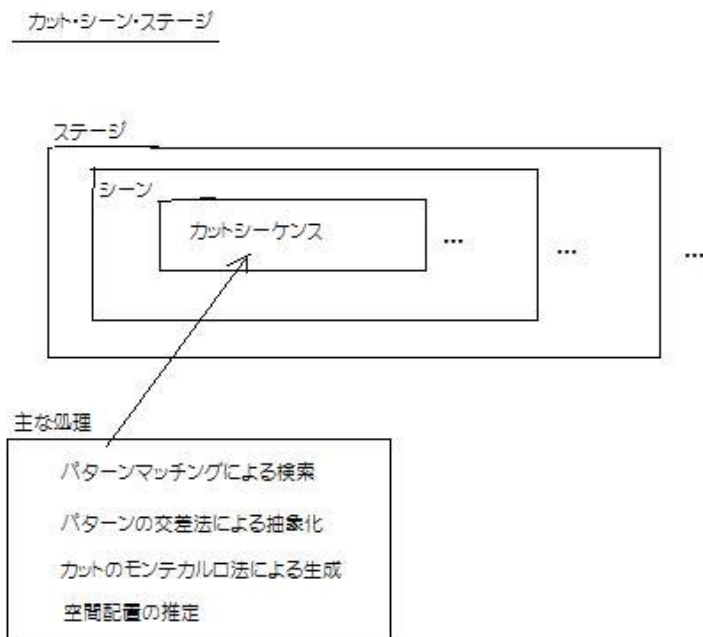
9. おわりに

評価システムを、その根本事象（センサー、アクチュエータ）群とデジタルツイン空間上での評価の連続の仕組みを考えてきました。そうして、デジタルツインを事象のネットワークとして生成していくためのプロダクションシステムを導入しました。プロダクションシステムは、パターンマッチングを行う条件を表現するコマンドと、それと連想する、デジタルツインのイメージをワークエリアに構築していくコマンドからなり、結果を評価しながら思考過程へと展開していくこと。それを説明しました。

最後に、平行処理の管理に焦点を当てて、コンピューティング環境の実現案を提案しました。

平行処理では、同時に利用するデータとかプロセス（プロダクションシステム）とかをセットで管理するのが良いように思われます。

データで説明しますと、データ（デジタルツインの部分）は次のような構造を持っています。データをセットで管理するとは、これらの構造のどの部分を同時に参照するかという問いになります。



カットの中は、多段階解像度でいくつもの階層を持って、点としての表現から、詳細な表現であるネットワークやマップへとブレイクダウンしていきます。

プロダクションシステムも、同時に動くプロダクションシステムはなにかということを経ダイナミックに管理するため、プロダクションシステムのセットによる管理機構を持つべきです。

プロダクションシステムのセットをチャンクとして纏めたものが直観になり、思考の基本機能に成ります。直感は、無意識的に動く思考の纏まり（チャンク）です。直感に対して、意識下で評価をダイナミックに実施しながら思考を実施していく、目標管理思考があります。目標管理思考は、直観をバンバン発行しながら、直観の結果を調停場というデジタルツイン空間に得て、調停場にあるデータを評価し、次の思考を組み立てていく過程です。

思考についてもう少し詳しく見てみます。

数学の問題を解く場面を考えて見ましょう。

[目標管理思考]

- (1) 試験用紙を見る。(Observe)
- (2) 問題の箇所を特定する。用紙の配置や内容を解析・認識するぞ(直観に任せる部分)
(Orient)
- (3) 問題を解く(「 $7 + 3 = ?$ 」)・・・計算するぞ(直観に任せる部分)(Decide)
- (4) 問題を解く(「 $? \dots > 10$ 」)・・・直観の結果を評価して問題用紙に書き留める。
(Act)

OODA ループを構成するのを見ます。

「直観」

- (1) 「7」、「3」を記憶の「+=」コマンドの引数に入れる。(Plan)
- (2) 「+=」のチャンク記憶から10を得る。(Do)
- (3) 結果は求める物かを評価する。(Check)
- (4) 最も良い物を得て調停場に納める。(Act)

PDCA ループを構成するのを見ます。

OODA ループも PDCA ループも基本的ステップで、評価という活動が挿入されているのを見ます。

おわり