

【要約】

We recommend object oriented technologies to construct AI(artificial intelligent) systems. We have been making several programs to research AI for several years, and found some object oriented technologies were suitable to resolve problems. They reduced the complexness of target concepts. We used RDB(relational data base) to hold data, and used them actually as object in memory. Our stance is to extract common functions from programs and to make them in object oriented technology modules.

1. はじめに

人工知能を実現するのに、どんな工夫が必要か、具体的にもものづくりする前に考えておくことは、作業の見通しを良くするためにも有効である。今まで、試行錯誤で作ってきた人工知能の経験を整理して、脳にも学びながら、基本的な技術と思えるものを記述して行こうと思う。

本論で目指すことは、オブジェクト指向技術による人工知能の実現である。データ（知識要素）の記録にRDBを用いる。RDBのレコードが1つの知識要素を構成するようにする。知識要素の内容は基本的にオントロジー記号の集まりで、それに重要度やタイムスタンプなどの付加情報が付く。RDBのテーブルはまとまった意味の塊である概念を表わし、本論ではモデルという名前と呼ぶことにする。モデルはJavaではオブジェクトということになる。そのオブジェクトも様々な機能モジュールで使いまわすために、基本的なインターフェースを設けて、そこから派生してつくっていくことになる。

本論の構成は、オブジェクト指向技術での人工知能の中心的なシステムを「基本技術」に記し、人工知能として基本となる機能群を「モジュール構成と処理概要」に記していく。

2. 基本技術

2. 1 オントロジーとモデル

人工知能の持っている身体とその外界を認識するのは、センサーの値を解析することで実現されることである。そのセンサーの値を構造をもって組織化して認識し、行動して行かなくてはならない。センサーと行動とをつなぐためにも、またシステムとしてあらかじめ組織化した機構を実現するためにも、センサーとアクチュエータと組織化したシステムとで基本となる約束事を決めていく必要がある。それがオントロジーであり、情報処理の基本データ形式としてオントロジーを構造化して記録する「モデル」である。モデルはオブジェクト指向で

いうオブジェクトである。

オントロジーは記号と値の対で表現する。そして、規格化して広く使われていくものである。

モデルは、オントロジーのセットであるが、構造としてリスト、ツリー、1次元マップ、2次元マップ、3次元マップという形にまとめられて使われる。

2. 2 重み付き投票法

人工知能は曖昧性のあるデータを扱うところに特徴がある。一つの事象からは、最終的な解釈は得られない。文脈や様々な処理結果の相互に影響し合った結果によって最終結果を求めて行かねばならない。その簡単な機構として、ニューロンネットワークに倣って、促進の伝達、抑制性の伝達の重要効果によって多数決で結果を出力していくというアーキテクチャを提唱する所以である。

とかく、投票法は解析プロセスの中に埋め込まれて実現されるが、本論では、解析などのプロセスとは独立したモジュールとして実現していくことを考えている。学習機構の装備や処理の見通しが良くなるし、組織だったシステムに仕立てて行けるからである。

投票法プロセスは確信度を保持するオブジェクト群のネットワークを管理する、確信度を保持するオブジェクトは実際の知識要素オブジェクトと相互に参照している。一部の知識が発火すると、それが参照している確信度を保持するネットワークオブジェクトを発火し、投票法プロセスは重み付けで接続しているネットワークオブジェクトを全部発火していく。発火はタイムスタンプを押していくことで実現する。同じタイムスタンプのネットワークオブジェクトの確信度は加算し、そうでなければ確信度をクリアし、指定の確信度を設定する。通常、一段だけの接続を発火するのみだが、特別な場合には接続の連鎖をして発火していく。

単語は品詞に分類される。この汎化という機構は、同じ接続をしていることを、知識要素のオントロジーをマッチングさせて、最大一致の知識要素を立てていくというアルゴリズムで実現する。

ネットワークの学習は、報償系の評価によって、有効ならば投票法の重みを増加し、間違った結果ならば重みを減らすということで行う。ネットワークの発火関係を調べて、ネットワークの多段が発火した場合には、ニューロコンピュータ技術でいうバックプロパゲーションアルゴリズムで重みづけを増減させる。

2. 3 連想と推論とパターンマッチング

知能は語られていないことを推論して補うことが主な機能である。知能指数はパターンから連想や推論を自動的に行って行って、適切な情報を知識から取り出していく能力である。

連想も知識要素間の関係であるから、オブジェクトデータ中に連想先の知識要素のポイン

タを持つのは良いが、これでは手作業によるポインタの作りこみをプログラミングしていかねばならなくなる。自律学習による連想関係の構築を考えた場合、知識要素群とは独立なデータシステムにしておくのが良い。また、知識要素の永続化ではデータはRDBにテキストで保存するので、ポインタは知識要素の名前であることが有利である。インメモリのオブジェクトに復元するときにはポインタであるから、その名前とポインタの変換テーブルをいずれはもたねばならない。テーブルは知識要素名付加的なテキストを付けた名前にしておくと、システムはすっきりするであろう。

推論は、連想のネットワークをたどって行って、条件があう知識要素群から値を取り出し、返却することである（ユニフィケーション）。

パターンマッチングは、オートマトンを基本としたアルゴリズムで実現する。知識要素のオントロジーセットとオートマトンの的に終端すればマッチングしたと判断する。

連想のオブジェクト配置は次の構成を取る。

(連想元オブジェクト) → (連想の属性) → (連想先オブジェクト)

2. 4 並行処理基盤（ハードウェア、ソフトウェア）

(1) ブロードキャストネットワーク

同じ入力データを複数のプロセスで解析する。また、データの意味を評価するとき、同じデータを複数の形、視点で解析し、結果としてまとめるということをする。プロセスは複数のPCに分散して処理するのが効率的である。特に、実時間処理を高速で行うには並行処理技術が欠かせない。そこで、同じデータを複数のプロセスに同時に渡すことを重要と考えれば、ネットワークはブロードキャストによって、一度にデータを渡し、プロセス側で自分に適用されるデータかをデータの識別子で自律的に判断して決定して処理するとした方が良い。それで、ブロードキャストを基盤とする、プロセスネットワークを提唱するものである。

(2) 検索と推論（Prolog 言語）

検索のキーとなるデータは構造を持っている。自然言語による検索は文を解析して、記号の羅列に落とせる。それは、名詞や形容詞、副詞、動詞などのオブジェクトと、オブジェクト間の関係である格によって表わされる、またオブジェクトの属性は修飾しているという情報配置関係記号で表わされる。これは、一階述語論理（もしくは簡単に Prolog 言語）に落とせて、検索はそのうちの変数へのユニフィケーション機能として実現できるはずである。

イメージについては、イメージを解析してマップ上の記号列として情報を得られるので、記号列であるオブジェクトとそのマップ上の配置関係である格情報によって一階述語論理に落とせるので、あとは自然言語処理と同じに扱える。また、格でなくてマップをそのまま使うのであれば、マップの配置情報を自律で解析する拡張的な論理で表現できるので、それで

検索と推論を行うと良い。

(3) S O A (Service-Oriented Architecture)

自然言語処理の解析プロセス群とか生成プロセス群,画像解析プロセス群などと,プロセスは機能のくくりによっていくつかのモジュールに分けられる.基本的にすべてのプロセスは並行処理にすべきであるが,このモジュールごとに異なるP Cに配置するということが行われてよい.P Cネットワーク,モジュール内ネットワークはバックボーンとなる通信機構があって,そこにプロセスが差し込まれて動作するというアーキテクチャが最も柔軟なものとなるはずである.それはすなわち,S O A (Service-Oriented Architecture) での人工知能システムの実現である.モジュールごとのローカルネットワークとモジュール間のグローバルネットワークがあることになる.並行処理はこのネットワークにプロセスを繋げるということで,実現していく.

3. モジュール構成とその処理概要

3. 1 基盤系

(1) オントロジー

2項タプル(記号,値)によってオントロジーを表わす.記号は更に,xxx#xxx#xxx というように,#で複数の英数字列を組み合わせた,構造をもったものをも表す.構造を成す記号は,アルファベット順に配置されていると,検索キーとして使うときは,全体の記号列が一意に決定できるので,高速な検索が可能であるである.オントロジーはJavaのHashMapの中に保存して利用していくことを想定している.

(2) 知識ベース(RDB,プライミング,インメモリーシステム管理,ニューロンシステム)

知識ベースはオントロジー列を,そのプライオリティ,タイムスタンプ,発火中かどうか,重要度などの情報を付加して,特別なオントロジーをキーとして検索できるようになっているべきである.だから,RDBで基本的なところを実現して,高速アクセスにはインコアのオブジェクトを検索していくという形で利用していく.このインコアオブジェクトは,検索に優先順位があって,文脈処理への対応を図っている.これをプライミングと呼ぶ.

(3) センサー

センサーの値はオントロジーにマップされ保持される.画像データや音声データは生なイメージだけでなく,構造を持ったイメージであるから,その解析結果も自動的に取得してモデルとして保持されるべきものである.

(4) アクチュエータ

行動は最後には身体を動かすコマンドに落とし込まれる.このコマンドをオントロジーとして規約を決めていくことが汎用の人工知能を創っていくのに重要である.さらに,コマンド列としてまとまった行動要素(「ものを掴む」といったレベル)もモデルとして保持していくことが重要である.

(5) ミラー系

他者の行動を自らの行動にマップしていく能力は,他者が何をしようとしているかを知るうえで重要である.また,物まねで行動を学習していくことにも有効な機能といえる.これは,画像データを身体行動データに対応させていくオントロジーがあって初めて可能となる.

ミラー系は画像パターンを解析してできたオブジェクトと自身の行動プロセスオブジェクトを相互連想させることである.相互連想により,自身の行動が外部からどう見えるかを推論していくことも可能となる.

(6) 世界マップ

身体のもの配置を管理するマップシステムである.重力を中心に管理していく.通常は,必要最小限の2組のオブジェクト群についてオブジェクト間の距離と方向情報を保存するテーブルであるが,自身から他の全てのオブジェクトの距離と方向情報をテーブルに持って良い.任意の2つのオブジェクトの位置関係を知るときには,3次元マップに展開して解析していくことになる.

(7) 自己・身体マップ

自分の身体の形,身体部品の配置を知る,あるいは管理するためのマップである.基本的にJava3Dの骨格の管理方法を想定している.つまり,オブジェクトとオブジェクト間を結ぶ関節の木構造の連鎖として自身の体を認識するのである.

3. 2 意志・評価系

(1) 評価機構

人工知能は自分自身を守ること,人をまもること,人を傷つけないこと,より効果的な行動結果を得ることなど,人工知能が存在している価値を評価していく,おもとの管理系(意志・評価系)が必要である.これによって,さまざまなセンサー情報から的確に行動をプランして,目的をもって存在していけるようになる.評価は様々のセンサーデータ,人工知能のモデル群から,そのモデル,オントロジーに連想している,評価基準知識ベースの値で,現在の状

況を示すオントロジー群に重み付け投票で評価していったら、もっとも大きな評価されたオントロジーによって行動を決定していくような仕組みである。

虎と一緒に檻に入れられていて、出口を見つけて逃走しようという状況を考えてみたい。「虎」は生命を脅かす存在である。虎を画像で捉え、その脅威を知識ベースで得る。そして、危険回避オントロジーを重み付け投票で発火していく。「虎」オブジェクトからの退避を行動コマンドとしてプランニングする。「逃げる」コマンドからの連想で、出口が無いか調べる。「出口を探す」コマンドがプランニングされる。これらの行動のとき、フォーカスも「虎」、 「出口」におかれる。自分が逃げる対象を「from」格とし、逃げる方向を「to」格としてコマンドは持つ。そうして、そういう一連の作業が統一して意志、評価システムにコマンドとして保持されているように機構を創っていく必要がある。

(2) プランニング機構

行動は小さな行動要素を組み合わせて、時系列に並べたものである。途中で割り込みが頻繁に起きるし、予測とは異なる状況が現出する。頻繁に行動要素は入れ替わるものなのである。それでも、最終行動コマンドは一つで、そのコマンドの下、モデル予測制御と割り込み処理が管理され、実現されていく。コマンドから行動要素列を作るのは、モンタージュ写真を創るようなロジックで行う。コマンドシーケンスが滑らかなように、かつ、もっとも目的が達成されやすいように重み付け投票で行動要素を選び、配置していくのである。

3. 3 機能系

(1) 学習機構

学習は重み付け投票システムの構築において行われる機能である。基本的に試行錯誤で重みを増減していき、もっとも確かな重みを創っていくことである。基本的に2項関係の重みであるが、論理的にはこのアルゴリズムはバックプロパゲーションと同じになる。

(例) やまんおかみ→山の神 : 「山」が「の」を投票 ; 「山」が「神」を投票

→山野上 : 「山野」の解釈に投票 ; 「山野」が「上」を投票

結果評価で確からしい場合 : 「山の神」の方の重み付け投票を強化する。

結果評価で誤っている場合 : 「山野」の解釈の投票をマイナスする。

重み付け投票システムは、学習を自律的に行う必要があるため、投票結果が有効なものかどうかを評価するシステムも随伴しているものである。

(2) 解析機構 (自然言語、画像、音声)

パターンマッチングを知識ベースに行い、マッチ度によって重み付け投票をしていき、最

終的な解釈に至る。これは、自然言語、画像認識、音声認識に共通のアルゴリズムで実現できる。センサーデータは入れ子構造をしていますので、マッチングパターンもモデルの入れ子（ツリー構造）をしてマッチングを深めていく。

解析プロセスは、基データが与えられたら自動的に実行していくものと、「意志」の下、試行錯誤で実行していくものがある。

(例) 空 → オブジェクトとして保持；空のイメージをオントロジーセットとして取り込む。

青い空 → 「青い」を「空」の属性として保持；青いのイメージをオントロジーセットとして取り込む。

青い空を見る → 「見る」を行動オブジェクトとして保持する。格「を」などの情報もよせて、見るイメージをオントロジーとして取り込む。「見る」の格関係やオブジェクトの取り方を構造を持ったデータ（フレーム）として保持する。

(3) 生成機構（自然言語、画像、音声）

フレームデータ群を基に新しいデータを生成していく。その内容を解析フィードバックして、手直しが必要ならば出力結果を変形していく。ここは生成変形文法のアルゴリズムが有効なものである。

(例) 「空を見る」イメージを浮かべる。

イメージをオントロジーセットに連想で落とし込む。

「見る」オントロジーセットとフレームと、「空」のオントロジーから単語と語順を連想して、吐き出す。「空を見る」との文ができる。

「空が青い」というイメージからオントロジーセットを連想で得る。「空」の名詞化ということで、「青い空」に文変形する。

最後に、「青い空を見た」という文を得る。

(4) 知能機構（推論、連想）

人工知能の全てに渡って連想による知識からのデータの補完作業が必要になる。これを統一的に実現するモジュールがこの知能機能である。連想は重み付きの結合度で表現され、これは重み付き投票法と同じ機構にもある。そんな、知識とは独立の連想データ機構が提唱される。また、この連想を変数へのユニフィケーションとして実現すれば、推論機構なる。

知能要素を持つデータは、基本的に、（動詞、格、名詞、格、名詞、・・・）と表わされる。

(5) 行動・思考機構

思考は行動と同じものである。アクチュエータを実際に動かさない行動であり、思考と行動はコマンドの連鎖で管理されるもので、コマンドの内部はプランニングで行動要素を集め

編集して実際にアクションを生成していくものである。行動結果は常に評価システムで評価されていくものである。

コマンドは基本的に、知識要素の形式を取る。コマンドを実際に実現するプロセス群は、予定の行動から得られるセンサー情報と現在の状態の差分から、特定の値変換してアクチュエータ値群を出力していくという形式を取る。プロセスもモデルとして表現されるものである。

(6) シミュレーション機構

ミラー系をもって、他者の行動・思考を推論したり、自分の今の行動から引き起こされる行動群の予測をしたりする機構である。行動と思考の環境制限値を意志・評価システムから発生させて、行動・思考モジュールを動かすことで実現する。

4. まとめ

オブジェクト指向によって、人工知能という複雑なシステムに見通しをつけることができた。情報は意味を持った塊である。作業の過程で発生するデータも、作業を行うプロセスも、永続化される知識要素もすべてオブジェクトとして捉えることにより実現していくとコンパクトないくつかのオブジェクトを扱うという形式でシステムを構成できる。このことにより、人工知能として重要な機能である、投票法や連想・推論が汎用なライブラリとして切り出して作りこむことができると示唆できたと考える。

まだ完全なものは創っていないが、これまで試行錯誤してものづくりに挑戦してきて、その結果をまとめることは有意義なことであると考え、論文にした。ここ論文執筆の過程で、いろいろものづくりしていた時には見えなかった、機構としての人工知能が分かってきたこともある。基本的に、モジュール分けしていき、共通のアルゴリズムはライブラリとして製品にしても良いようなレベルの抽象化がなされると良いと考える。そこを踏まえて、詳しく論理構造を記したつもりである。このようなライブラリを人工知能の共通の道具として速やかに構築されることを願う。

おわり