認知機構の一方式　　2013/09/02（月）

人工知能をプログラミングしていくと、様々なデータが現れ、それらが連想関係で結ばれます。これが結構面倒なコーディングになっていきます。データの管理をどうするかとか、連想をポンタで張ると、検索が大変とか。こう、見通しが悪くなるのです。プログラムの一部分から見ると、コーディング世界はジャングルの中で、全体が見えなくなるのです。こう、なんかもどかしい。人間の思考は全体を見通して行われるように感じますから、この辺・・・プログラミングには何か工夫が必要かなと思うわけです。

そこで、自分の頭の中の思考を内省してみました。次のような特徴がありました。

（1） なんでもオブジェクトにできる。
（2） データの動的型変換がなされる。
（3） 場（シーンを見通した）を持った認識・検索・パターン発見・パターンマッチングがなされる。
（4） 何でもオブジェクト操作で処理が実現される。オブジェクトに、どんな操作もできる。
（5） 変化のトレースを覚えている。
（6） 色々なオブジェクトを「覚えていて」(同時に見ながら)、いつでも、部分オブジェクトを「観ること」ができる。
（7） オブジェクトのどの部分も瞬時に検索・パターン発見・パターンマッチング（「観ること」）ができる。

こんなことができるアーキテクチャってどんなものが考えられるでしょうか。

1．　マップシステム

「場」は１次元、２次元、３次元のマップではないでしょうか。そのマップの点にオブジェクトが置かれます。検索はマップの点を高速に走査して行っていく。点は複数のオブジェクトが置かれ、交差とか内包、外苑、配置関係とかが高速に解析できるものです。オブジェクトが自動型変換ができるものであれば、オブジェクトの型を気にせず、点に配置し、操作できます。

行動を起こすコマンドは、記号システムですから、マップシステムを解析して記号列を得なくてはなりません。記号は、オブジェクト名と意味・属性オントロジー群、格情報（これもオントロジー）からなります。「連想」も「修飾」も格で表現しましょう。

マップ自身もオブジェクトです。なんでもオブジェクトとすれば、常にフラットで、全体を見通せて処理するプログラム環境となるでしょう。そんなハードウェアを工夫すれば（マップシステムをＧＰＵで処理するとか)、高速な認知システムが構築できるでしょう。

２．　プログラミング

（１）　基本的にデータの自動型変換ができることが重要です。自動型変換できない言語を使ったコーディングでは、自動型変換を自前でやっていくことになります。オブジェクトにそんな機能をコーディングして持つことですね。

（２）　マップシステムを作って、処理をコーディングしていくこと。マップシステムのハードウェアを使えるならば、それとのインターフェースを持つこと。

（３）　コマンドシステムを作って、行動・操作を実現すること。オブジェクト名、オントロジー記号列、格の羅列で、コマンドを設計します。プロダクション・システムも Prolog システムもマップシステムで管理しましょう。

３．　ハードウェア

（１）マップシステムの検索処理は全てハードウェア（ＧＰＵかも）で実現します。パターン発見、パターンマッチングも頑張ればハードウェアで実現できるかもしれません。パターン発見もパターンマッチングも２項関係だから、かなり微細な処理単位の結合で、実現できると考えられるわけです。

おわり