

信濃の大局的なモジュール構成を記します。

1. データベースとインメモリデータ

- (1) 信濃のデータはRDBで持ちます。そのRDBの一つのレコードをインメモリにしたオブジェクトが **FrameData** で、管理データ以外の実質の人工知能データはその一部のオブジェクトで、**Node** として持ちます。
- (2) **Node** の集まりを **NodeSet** オブジェクトで管理します。
- (3) **FrameData** と **Node** と **NodeSet** を管理するのが知識ベースモジュールです。

2. 領野管理

同じタイプのプロセスが並行処理として複数実行される単位が領野であり、同じタイプのデータが集まった単位も領野として管理すると見通しがよいものになります。

- (1) 領野オブジェクトは **Field** オブジェクトです。
- (2) **Field** の集まりを管理するオブジェクトが **FieldSet** で、**FieldSet** の集まりが **FieldSetCluster** です。
- (3) プロダクションシステムの領野を特に、**ProductionSystemField** オブジェクトとします。その集合が、**ProductionSystemSet** です。
- (4) コマンドの領野は **CommandField** とします。その集合が **CommandFieldSet** です。
- (5) 部品的な領野はコマンドとプロダクションシステムで、パターンマッチングや検索、連想（重み付投票）を行います。

色々のタイプのパターンセットに対して、パターンマッチング並行処理の塊を構成します。このパターンセットは沢山あるから、パターンマッチング領野は多数存在するでしょう。

連想処理である重み付き投票は、ノードの発火を実現する機構です。発火は全ての領野のノード間に対して行われることですから、大きな一つの領野になるでしょう。発火はフォーカスすなわち、発火の重要度管理（プライミング）として実現されます。重要度の高い順に発火したノードを並べて登録することが必要ですので、・・・しかも、領野毎に重要度を管理しなくてはなりません・・・のどのある領野毎に、プライミングするプロセスが並行で動作しなくてはなりません。この辺、ハードウェアとソフトウェアでの工夫が必要です。アレイリストでできるだけロックをしないでアクセスできるようにするとか、領野の中に、複数のサ

ブ領野を持って、共有メモリにするとかです。

- (6) 総合的機能を実現する領野は、類推、思考、自然言語処理、画像認識処理、意志・規範システム、機械学習システムです。
- (7) 領野には基本的にフォーカス機構が装備されていて、データの優先度を管理し、合目的な処理になるように制御していきます。
- (8) 領野内の基本データオブジェクトは **Node** で、脳のコラムに対応します。

3. コンセプト

データの基本はオントロジーとなる記号と値ですが、機械学習で獲得したデータはコンセプトとして管理します。コンセプトは領野を渡って、**Node** を集めたものが成ります。

おわり