知識ベースへのプラグインの一つの例として日本語処理を考えてみたい。日本語処理としては次のような機能がある。

（１）文章認識
（２）問い合わせ回答
（３）文章生成
（４）イメージ（ボクセルデータ表現）をパターン認識して意味記号化する
（５）意味記号からイメージ（僕セルデータ表現）を生成する

１．文章認識

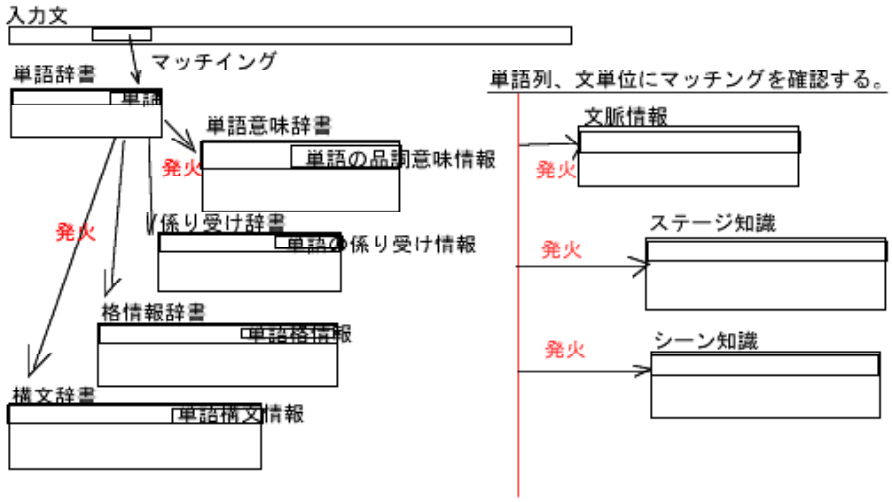

図　文章認識概要図

文章認識も「発火」を基本に構築していく。長文や長い文章を解析するときのメモリ制約や処理時間制約を考えて、文頭から逐次、文脈やステージ、シーン知識と照合し、現在の状況を実時間で把握しながら文章を解析していく。できるだけ早い段階で、文意を決定していく。

最終的にはコーパスモデルを作っていく事になる。知識の取り出しはコーパスからのオンデマンド解析によって必要に応じて行っていくものとする。カテゴリーなどはこのオンデマンド解析で充実していくことになる。また、ステージやシーン、アクターのモデル知識の充実もオンデマンドでやっていくことにする。

文脈は、欠落語や指示代名詞の同定、表現の微妙な違いの対応に必要な知識となる。文脈はカレントの知識、アクティブ（発火時点が近い）な知識であるから、他の知識より優先されるのである。

【API】

```
Module sentenceRecognizer=new SentenceRecognizer(AI ai);
                                        //文章認識プラグインモジュールを定義する
```

文章認識プラグインの中では、以下のモジュールを使っていくことになる。

```
SentenceAnalizer sa=new SentenceAnalizer(AI ai);//１文解析モジュールの設定
```

FrameData には、文章も XML 形式で入っているので、それを１文ずつ取り出す必要がある。

```
Iterator sitr=fd.getSentences();
while(sitr.hasNext()){
    String sentence=sit.next();
    FrameData fd=sa.analize(String sentence)//１文を解析する
}
```

２．問い合わせ回答

学校に行って、校庭で遊んでいたよね。A 君に何貰ったの？

というような文章では、「学校に行って」「校庭で遊ぶ」が文脈構築である。この文脈から過去のステージ、シーンモデルを発火させて、答えに結びつくコーパス文章・・・知識部分を発火させることが出来なくてはならない。知識の中に、
「A 君にボールをぶつけられた」「A 君は私にハンカチを渡してくれた」などの知識があれば、「渡す」＝「貰う」の方向違い・・・というような知識によって、「A 君は私にハンカチを渡してくれた」を選択し、そこから知識を抽出して、「A 君にハンカチを貰った」という文を組み立てて表出する。

相手の話の内容、自分が話した事から、相手が既に知っている事を推定しなくてはならない。これは、モデルの持っている知識を利用して推定したり、話題の枠組み（ステージ、シーン、話題フレームなど）の持っている情報で、モデルが持っていない情報として推定したりする。自分が知っていることを５Ｗ１Ｈ１Ｆで捕らえて、その中で重要な事から回答していくようにする。相手の問いに答えることもある。

このように、問い合わせ回答でも、「発火」と基本的な知識加工プロセスがあって実現できる事がわかる。知識加工プロセスは学習によって充実していくべきで、コマンドパターンからの実行が可能なプロセスも作り込む必要がある。コマンドパターンは思考システムで生成していくことになる。

【API】

Module quetion_answer=new QuetionAnswer(AI　ai);
//問い合わせ回答プラグインモジュールを定義する
quetion_anser.execute(FrameData quetion);//問い合わせ実行
//問い合わせの結果は意志システムの待ち合わせに回答として保存する（問い FrameData に結果として回答 FrameData を連想させる）

３．文章生成

知識要素に振られている重要度の高い物を選んで文章化していく。また、文脈に登っている知識を優先に文章化していく。文章の意味列か知識記号を単語に変換したり、文法、構文を連想発火して取り出し、シームレスになるようにシステミック文法で繕って、最終文章の形を決定し、表出していく。

発火する単語や文法、構文パターンなどは複数存在するが、好みとか文脈というものでフィルターしていくことで、最終的に一つを選別していく。

システミック文法としては、

・重要事項を先に置く
・時間、場所、事象の順に言葉を紡ぐ
・主題は助詞「は」を用いる
・疑問は「かもしれない」を用いる

などなどである。係り受け、係り結びに注意して単語列を決定していかねばならない。

発話の順序もパターンとして知識に持っていることになる。文体などもこの発話パターン知識である。

【API】

Module sentence_generator=new SentenceGenerator(AI　ai);
//文章生成プラグインモジュールを定義する
sentence_genertor.execute(FrameData　fd);//FrameData　fd は文生成の種になる知識要素。ここから連想によってどんどん物語を広げていく事になる

４．イメージをパターン認識して意味記号化する。

ステージ、シーンを解析していかねばならない。また、物の形状とか位置関係を解析していかねばならない。イメージの全ての状況、情報を記号としてもつのは不可能である。情報は無限にあり、視点によって情報は異なる記号で表現されもするからである。「与える」と「貰う」の関係とか、「右」と「左」の関係とかいろいろ視点のもつ問題はある。これらはオンデマンドでイメージを解析くることによって記号化し、文脈によって記号発火されねばならない。

５．意味記号をイメージ化する。

視点変換の為に、いったん意味記号をイメージ化しなくてはならない。

イメージ構成は、イメージ要素をアセンブラしていく過程でもある。また、イメージはロボットが生活していく中でコーパスモデルとして取り込んでいく視覚画像でもある。２次元イメージを３次元イメージ（ボクセル）に構成して、知識化していく過程も学習システムにあるということである。

６．イメージの永続化表現（ボクセル）

イメージは３次元ピクセルセット（ボクセル）で表現し、ＸＭＬ形式で永続化していく。各点は座標と、様々な付加情報とを持つ。次のようになるだろう。

```
<image>
 <point>
  <x>10</x>
  <y>10</y>
  <z>10</z>
  <attribute>
   <color>
    <r>5</r>
    <g>5</g>
    <b>5</b>
   </color>
  </attribute>
  <object>
    <head/>        ・・・・・「頭」という情報を付加する
    <hair/>        ・・・・・「髪の毛」という情報を付加する
  </object>
 </point>
 <point>
  <x>11</x>
  <y>10</y>
  <z>10</z>
  <attribute>
   <color>
    <r>5</r>
    <g>5</g>
    <b>5</b>
   </color>
  </attribute>
  <object>
```

```
      <head/>
      <hair/>
   </object>
  </point>
  ・・・
</image>
```

サブイメージから構成されているときは、<subimage>・・・</subimage>の間に、イメージの URL(レコード識別子 AT テーブル名)で指定する。

属性が別レコードに指定されていれば、<attachedattribute>・・・</attachedattribute>の間に属性定義レコードのＵＲＬ（レコード識別子 AT テーブル名）で指定する。

７．知識の分類

知識要素にはタイプがある。それらはつぎのようなものである。

（１）事実
（２）体験
（３）空想、架空の事
（４）間違い
（５）推測
（６）知っている事
（７）知らない事
（８）知りたい事
（９）好みな事
（１０）嫌いな事
（１１）危険な事
（１２）安心な事
（１３）勧める事
（１４）避けるべき事

８．学習したコマンドの利用

作りつけのアルゴリズム（ハードコーディングされたメソッド）でなくて、操作を学習して、それをコマンドパターンで実行したいことがある。それには、コマンド操作の場とコマンドを汎用的に処理するメソッドが必要になる。

場を提供するクラスを BlackBord とし、コマンド(FrameData で表現する）をこの BlackBord の特定のメソッドで実行するようにする。この BlackBord に格人工知能の作業データを持ち込むのは格作業モジュールの責任である。BlackBord は数学的な能力を実現する作業しかできないのである。

９．おわりに

文章認識も、文章生成も、イメージをパターン認識して記号を得ていく過程も、意味記号からイメージを得ていくことも、学習システムによって能力を充実していくことが必要なため、コマンドパターンで動くプロセスを設計しておく必要がある。そうして、コマンドパターンを得ていく過程は思考システムである。思考システムをトレースするデータをコーパスとして収集していき、それを抽象化して、コマンドパターンにしていくのである。

プログラム作成に当たって考慮しておきたい事は次のような事柄である。

知識処理は、

センサ（オントロジー）->コーパス->モデル->カテゴリー

という風に知識を構築していくのが素直である。これは知識のセントラルドグマと呼べるかもしれない。来れに従えば、学習システムの設計の見通しが良くなるのである。

学習は連想をどう築き上げるかという問題が全てであると考えられる。パターンも構造との連想であるし、単語も記号との連想である。連想は共起解析の結果であるから、共起解析をスマートに作り込めれば、ほとんどの学習機構は成ったも同じであると言える。

言葉の学習は、

（１）単語切り出し
（２）品詞推定
（３）格関係の推定
（４）係り受け推定
（５）意味推定

をしていかねばならないが、その中で、どうしてもイメージデータの解析をして、そのイメージデータから得られる構造を持って､格とか意味の推定をしていかねば成らなくなる。つまり、学習系は前提として、

（１）パターン認識能力
（２）固まり把握とフォーカス能力
（３）対応関係構築能力（共起解析）

の３つの能力は必須となる。

おわり