

意味談話理解のための知識処理

木下 聡・浮田輝彦・住田一男・天野真家
鶴東芝 総合研究所

1. はじめに

我々は意味談話理解の研究の一環としてフレームとルールマルチパラダイムの知識表現を基本とするコンサルテーションシステム (Information Service System by Analyzing Conversational Context - ISAC-) を開発しており、現在はタスクとしてビデオの操作法に関するコンサルテーションを行っている[1]。本稿では同システムにおいて意味談話理解とタスク処理で共用される知識の表現と知識処理系について報告する。

2. 知識表現

2. 1 対象分野の知識の表現

対話システムにおいて、計算機がユーザ（対話の相手）の述べることを正しく理解するには、現在の対話の状況と対象分野に関する知識が必要なことは明らかであり、ユーザの表現が、より文脈に依存したものになるにつれ、知識に対する重要性がより大きくなる。ISAACでは、対象分野に関する知識を、意味談話解析とタスク処理の2つで共用できるものとして表現している。当システムでは、知識をスキーマとルールの2種類の表現型式で表現しているが（図1）、内容的には以下の2つに分類される。

1. 語に対応する概念に関する知識。
2. 操作方法など手続きに関する知識。

概念に関する知識としては「ビデオ」や「テレビ」など「もの（以下ではオブジェクトと呼ぶ）」に関する知識と、「押す」、「再生する」といった動詞など「こと」に関する知識に分けられる。我々はこれらの知識をスキーマと呼ばれる枠組で表現している。

スキーマは図2に例を示すように、フレーム部とルール部からなっている。フレーム部では、概念の上位-下位関係や部分-全体関係、オブジェクトの属性などが記述される。また、ルール部にはオブジェクトの動作や属性間の論理的关系がif-then ルールとして記述される。さらに、動詞など“こと型”の概念も種々の格をスロットとするスキーマで表現される。

自然言語の特徴の1つとして、表現の多用性がある。たとえば、現在機械の電源が入っていることを表現するにも「電源はONである」、「電源は入れた」、「電源スイッチは押した(推論により分る)」などの表現が可能である。当システムでは、これらの表現の等価性は、オブジェクトのスロットにおける表現の等価性として扱っている。すなわち、それらの評価の結果、スロットとしての表現が同じになれば等価な表現と見なしている。

手続きに関する知識は、本システムでは、操作の手順となる動作の列をif部、操作の目的(結果)をthen部とす

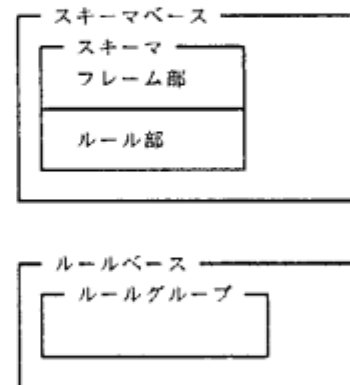


図1 知識ベースの構成

```

schema(cls.vir
  [(superC, [(value, [電気製品])]), % スキーマのフレーム部
    (has-part, [(value, [電源スイッチ,
      (再生スイッチ, ...)])]),
    (ステイタス,
      [(car-num, 1),
        (value-class, integer),
        (value-condition, (-V, (V>=1.V<12)))],
        (電源, [(enumeration, [on, off])]) ]),
  [s-rule(dengen, dengen-on=1.V, % スキーマのルール部
    (kr-schema(V, [(電源, [off])], true), % 条件
      event(押す, 1, [(object, [SV])]), % 条件
      kr-schema(V, [(電源, [on])], true), % 結論
      [(('$var-constraint', (
        V = v i r :
          SV # 電源スイッチ :- part-of(SV, V)
        ))) ] ]),

```

```
rule(ビデオ操作, カセット取り出し,
  event(押す .1, [(object.[SW3])]),
  event(出る .1, [(object.[C]).(source.[V])]),
  [(('$var-constraint'-(
    SWスイッチ=part-of(SW-V):
    C=カセット:
    V=video)))])
```

図2 知識表現の例
(一部簡略化してある。また“%”以降はコメントを、
大文字で始まる単語は変数を意味する)

るif-then ルールとして記述している。タスク処理において操作法の問合わせを受けた場合には、then部によりルールを検索し、if部を操作手順として得る。また、ユーザが「～をした」といった操作を行なったことを述べた場合には、その動作をif部の要素として含むルールを評価し、状況の変化を推論することも可能である。これらの知識は、談話処理において、照応の候補が複数個得られた時に、最適な候補を決定するための情報として使用されている[2]。

2. 2 ワールドを用いた推論

当システムにおいては、入力文の解析によって生成されるインスタンス・スキーマやルールの実行により追加されるデータは、テンポラリなデータとしてWMに記憶される。WMの内容は対話を進めるにつれ、次々に変更されていく。そこで、タスク処理における仮説の取りや談話処理における同定の取りやに対処するための手段としてWMをtree構造として、履歴を保持している。ここではこれを「ワールド」と呼ぶ（KEE などではcontext と呼ばれているものに相当する）。以下ではワールドを用いた処理の一例を示す。

(1) 談話処理

談話処理においてはまず、新しい入力に対し現在のワールドの子供として新しいワールドを生成する。そのワールド上で入力文中の名詞に対応する概念のインスタンス・スキーマを生成し、既に存在するスキーマとの比較や、他の知識を用いて照応参照処理を行う。処理の結果既に存在するスキーマと同定された場合には、削除される。また、照応の候補を決定した後、さらに前向き推論などを行って、文脈との関係をより詳しくチェックすることも可能である。

(2) タスク処理

タスク処理においては、ワールドを用いた推論を現在とは主として条件文（「もし～したらどうなるか」）の処理に使用している。このような入力に対しては、(1)と同様に、まず新しいワールドを作り、そのワールドにおいて条件文の仮定の部分を事実として評価し、それから推論される事象を回答している。

3. 問題点

ここでは対話によってユーザと協調的にタスク処理を行って行くシステムにおける問題点を述べる。

(1) クラスの同定について

クラスAの下に、サブクラスとしてB, C, Dが定義されているとしよう（図3）。入力文中にクラスAに対応する名詞Waが現れたとき、A（およびそのすべての下位クラス）のインスタンスとしてインスタンスA_iが既に存在し、さらに意味談話解析の結果としてWaがA_iと同定できる場合には問題ない。そうでなければ、Waは今現在システムが知っているAのインスタンスとは別のオブジェクトであると考えられ、新たにAのインスタンスとしてインスタンスA_jを作る必要がある。しかし、タスク処理によっては、下位のクラスのいずれであるかを決定していく必要が生じる。

(2) 構成要素の段階的生成について

一般に幾つかの構成要素をもつ有形物の下位概念Aを考えた場合、Aのインスタンスは基本的に、Aのhas-partに記述されている各構成要素のインスタンスを構成要素とし

て持つと考えることができる（図4）。例えば現在あるビデオの知識表現においては、ビデオの構成要素として約80の構成要素がある。このような知識があるとき、インスタンスを生成する度に、一度にすべての構成要素のインスタンスを生成し、has-part階層を作るのは、資源や処理時間の点から満足できるものではない。そこで、意味談話解析やタスク処理における必要に応じて、段階的にインスタンスを生成していく必要があると考えられる。

4. おわりに

ISACにおける知識の表現と現在の問題点について述べた。今後、現在のシステムにおいてさらに実験を行い、知識の表現と推論機能などについて評価・検討を行っていく予定である。なお、本研究は新世代コンピュータ技術開発機構（ICOT）の委託により行っている。

(参考文献)

1. 浮田他, "談話理解機能を持つ機器操作案内システム概要", 情報処理学会第85回全国大会予稿集, 1987
2. 住田他, "最適解釈判定法による会話文の理解", 電子通信学会NLC研究会 NLC87-16, 1987

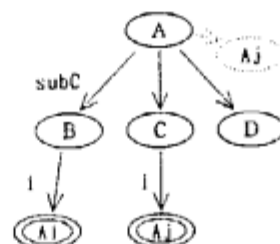


図3 クラスの同定の問題
(○はクラス, ⊙はインスタンスを表す)

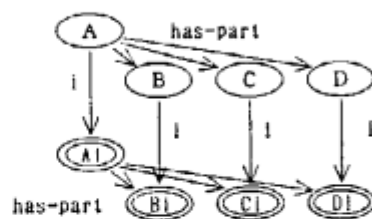


図4 構成要素のインスタンスの生成