マニュアルレスシステムにおける説明文生成(2)

5J-2　槇本英治　垣内隆志　上原邦昭　豊田順一
大阪大学産業科学研究所

1 はじめに

本報告では、前稿[2]で述べたマニュアルレスシステムのうちユーザモデル、プラン修正部、プラン洗練部、および文生成部について説明する。

2 ユーザモデル

ユーザの知識のレベルに合わせた説明を行うために、ユーザとの対話過程からユーザが知っていると結論づけられる事柄を事実として記録し、その記録をユーザモデルとして利用する。たとえば、

user: テキストエディタとは何ですか?

system: テキストエディタはプログラムを記述したテキストファイルを編集するプログラムである。テキストエディタには行エディタや画面エディタがある。

という対話の後では、以下のような事実が記録される。

know-of(user,テキストエディタ).

このユーザモデルの変更により、システムはユーザがテキストエディタについて十分な知識を持っていることを推論する。この結果、以後の説明文でテキストエディタについて詳しい説明が行われることはない。

ユーザモデルには、対話によって得られた事実として記録されていないが、既に得られた事実から真である事柄を推論するためのスキーマも含まれる。以下にスキーマの例を示す。

ユーザがディレクトリの構造とディレクトリを操作するコマンドを知っている
→ ユーザはどのファイルについても、その場所を知っているか、あるいは探索により知ることができる

上記のようなスキーマは、人間が常識として持っているものが多く、現在はその一部分のみを形式的に記述している。

3 プラン修正部

図1に、ゴール'DEFINE(アセンブリ言語)'を展開したプランの木構造を示す。
ここでは、以下のようなプラン列が得られている。

P4, P6, P14, P16, P8

プラン修正部は、探索木が一段深くなるごとに起動され、新しく付加されたプランが以下の2つの条件を満たしているかどうかの評価を行い、条件を満たしていないものを削除する。

1) プランが概念構造を上方向にたどらない
2) プランの重要度が臨界値0より大きい

条件1は、上位概念をたどることにより説明文の話題が全く関連のない概念へと移ってしまうことを防ぐものである。

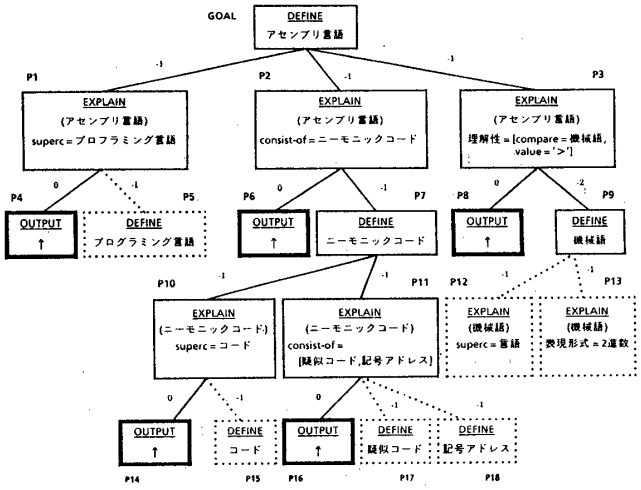

図1 展開終了後のプラン

プランP5, P15が破線で囲まれているのは、その親ゴールP1およびP10が上位概念についての説明を行うプランであるために削除されたことを示している。

「プランの木構造」中に現れる各プランには、それぞれ重要度と呼ばれる値が与えられる。プランに与えられた重要度は、そのプランがプラン列中でどの程度重要であるかを表している。「プランの木構造」の根に対応する初期ゴールには重要度4が与えられ、各プランの重要度は下位レベルに至るに従って小さくなる。

条件2は、プランの木構造が深くなりすぎて話題の中心から離れるのを防ぎ、ユーザの既知概念についての冗長な説明を防ぐものである。あるしきい値より小さな重要度を持つプランは、話題の中心から離れているとして、条件2により削除される。このしきい値を臨界値と呼ぶ。たとえば、図1中のプランP17, P18は重要度が臨界値0以下になり、この条件を満たしていないために削除されている。

プランに与えられる重要度は、初期ゴールの重要度と、そのプランに至るまでの各枝につけられた重みとの和として計算される。枝の重みは、ゴールの展開に用いられたオペレータ中のプランに付随して記述されている。

新しい概念についてユーザが十分な知識を持っている場合には、説明を簡単にすまさなければならない。新しい概念の説明をユーザの知識に応じて詳しくしたり簡単にしたりするため、DEFINE型のプラン[3]の枝には重みとしてアルファベットVで始まる変数がつけられている。ユーザモデルにより、ユーザが説明される概念に関する知識を持っていると推論されるときはこれらの変数に-2が、そうでないときは-1が代入される。この枝の重みの違いによって、ユーザがよく知っている概念について説明を行うプランの重要度が相対的

Text Generation of an Online Help Facility (2)

Eiji MAKIMOTO, Takashi KAKIUCHI, Kuniaki UEHARA, Jun'ichi TOYODA
The Institute of Scientific and Industrial Research, Osaka University

に小さくなり、不必要な説明を省略することができる。たとえば、図1中のプランP9への枝に-2がつけられP7への枝に-1がつけられているのは、ユーザは機械語について十分に知っており、ニーモニックコードについてあまり知らないことが、ユーザモデルより推論されるためである。この結果、図1中のプランP12, P13は、重要度が臨界値以下になり条件2を満たしていないために削除されている。

4 プラン洗練部

説明文のような文章は、通常ある特定の概念について書かれているものである。しかし、文章中の個々の文について着目すると、それらは常に同一の事物を言及しているわけではない。文章中に現れた新しい概念について次々と説明を加えたり、もとの話題に戻ったりしながら文章は展開していくものと考えられる。このように、文章中で現在話題の中心となっているものを焦点と呼ぶ。

プラン洗練部は、Text Planningによって得られたプラン列から、話題の流れが明確な文章を生成するために、以下の3つのデータ構造を用いて焦点の移動を管理している[3]。

cf (current focus) 現在、焦点の当たっている概念
pfl (potential focus list) 焦点の候補となる概念のリスト
fs (focus stack) 以前の焦点を蓄えるためのスタック

以下、図1のプラン列から説明文を生成する際に、これらのデータ構造がどのように用いられるかについて説明する。まず、プランP4が「アセンブリ言語」についての記述であるため、cfには「アセンブリ言語」がセットされる。次の文で焦点の当たる可能性のある候補「プログラミング言語」がpflにセットされる。また、fsには初期値として[]がセットされる。次のプランP6も同様に、「アセンブリ言語」に関する記述であるため、焦点は移動しない。この結果、重文化処理が行われ、プランP4, P6から以下の文が生成される。

「アセンブリ言語はプログラミング言語の一種であり、ニーモニックコードから成る。」

またpflには[ニーモニックコード]がセットされる。プランP14, P16はpfl中の要素「ニーモニックコード」についての記述であるために焦点がシフトされ、前文の焦点である「アセンブリ言語」はfsにプッシュされる。この結果、

「ニーモニックコードはコードの一種であり、疑似コードと記号アドレスから成る。」

という文が生成される。プランP8で再び「アセンブリ言語」について記述されるため、fsに蓄えられた「アセンブリ言語」がポップされ、以前焦点の当たっていた概念に焦点が戻る。この結果、文章の流れが変わることを示すために段落分けが行われ、最終的に以下の説明文が生成される。代名詞化、主語化処理についての説明は別稿[1]に譲る。

「アセンブリ言語はプログラミング言語の一種であり、ニーモニックコードから成る。ニーモニックコードはコードの一種であり、疑似コードと記号アドレスから成る。

アセンブリ言語は機械語と比較して理解性が高い。アセンブリ言語で書かれたプログラムはアセンブラによって機械語に変換される。」

5 文生成部

プランから文を生成する際には、まず機能構造(Functional Structure)と呼ばれる文の内部表現に変換される。機能構造は属性と値の組の集合で表され、部分機能構造を属性の値として許しているために、階層性を持つ構造を表現することができる。

プラン中の記述を表層化する場合には、プランと機能構造の組からなるテンプレートによって機能構造に変換される。たとえば、図1中のプランP6は図2の機能構造に変換される。

```
[主格 = [主要部 = [述語 = アセンブリ言語 ]]
対格 = [主要部 = [述語 = ニーモニックコード ]]
述語 = 成る(↑主格, ↑対格)                    ]
```

図2 機能構造

本システムには、機能構造中の各意味述語に対してFS(Functional Structure)と呼ばれる構造が予め辞書形式で与えられている。FSとは、述語の必須格に対して格助詞を付与するときの骨組みとなるものである。FSは必須格の名詞句と述語から構成され、以下の3つのうちいずれかの形式をとる。

一つの無標の名詞句と共起する述語のFS
[無標の名詞句 述語]
二つの無標の名詞句と共起する述語のFS
[無標の名詞句 [無標の名詞句 述語]]
三つの無標の名詞句と共起する述語のFS
[無標の名詞句 [無標の名詞句 [無標の名詞句 述語]]]

機能構造が与えられると、まず述語の辞書中からFSが取り出され、格助詞の付与が行われる。任意格の名詞句については、その名詞句と述語との格関係から格助詞が決定される。有標の名詞句に付与される格助詞は、各述語によって一意に定まっているため、述語辞書中のFSにあらかじめ記述されている。無標の名詞句には、格助詞の付与規則によって格助詞が付与される。たとえば、図2の機能構造から文を生成する場合、機能構造から以下のFSが構成される。

[_____ [___から___ なる]]
アセンブリ言語 ニーモニックコード

さらに、このFSに対し付与規則を適用した結果、FSは

[___が___ [___から___ なる]]
アセンブリ言語 ニーモニックコード

となる。

格助詞付与の後、必須格の名詞句はFSに現れる順序に従って並べられ、以下の文が得られる。

「アセンブリ言語がニーモニックコードからなる。」

6 まとめ

本システムは、説明文のプランニングにおいてユーザモデルを用いてプランの重要度を決定するため、ユーザの理解度に応じた柔軟な説明文を生成することが可能となっている。現在のユーザモデルは、簡単な事実の集合と少数のスキーマより成っている。今後はスキーマを増やすことにより、ユーザの知識レベルをできるだけ正しく近似したユーザモデルを実現することが必要である。

参考文献

[1] 垣内他:焦点の概念を導入したマニュアルレスシステムの説明機能について, *Proc. of the Logic Programming Conference '86*, pp.19-26 (1986).

[2] 垣内他:マニュアルレスシステムにおける説明文生成(1), 第33回情処全大, 5J-1 (1986).

[3] Sidner, C. L.: Focusing in the Comprehension of Definite Anaphora, in M. Brady et al. (eds.), *Computational Models of Discourse*, pp.267-330, MIT Press (1983).