

マニュアルレスシステムにおける説明文生成(1)

5J-1

垣内隆志 榎本英治 上原邦昭 豊田順一

大阪大学 産業科学研究所

1 はじめに

現在、マンマシンインターフェース機能を高度化するために、システムの説明機能を充実させようという研究が盛んに行われている。たとえば、UNIXのコマンド体系についての質問に答えるUC^[2]には、ユーザの熟練度に応じて説明を変えるために、簡単なユーザモデルが導入されている。しかしながら、ユーザモデルからのフィードバックが説明文に十分反映されておらず、生成される説明文は「文章としてのつながり」が全く考慮されていない。また、データベースの構造に関する質問に答えるTEXTシステム^[1]は、談話構造の定型的な枠組み(スキーマ)を用いて文章構成を決定しており、つながりのある文章を生成することが可能である。しかしながら、文章構成はスキーマによって予め定められているために、ユーザの理解に応じて説明を修正することができない。

以上のような観点から、我々はプランニングの手法を用いて、ユーザの理解度に応じた説明文を生成するマニュアルレスシステムを開発している。

2 システム概要

図1に本システムのシステム構成を示す。

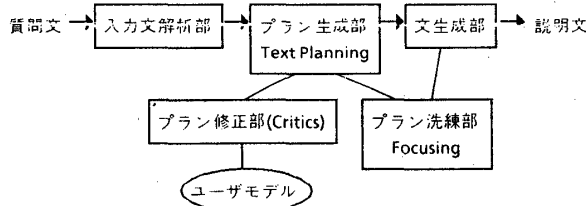


図1 システム構成

説明文を生成する際には、まず何をどのような順序で述べるかを決定しなければならない。この過程を、ユーザにある事柄を理解させるとい目標(ゴール)を達成するための手段(プラン)を推論する過程とみなすと、一種のプランニング過程として捉えることができる。このようなプランニングによって説明内容および文章構成を決定する処理をText Planningと呼ぶ。プラン生成部は、抽象レベルから具体レベルへと段階的に詳細化を行う、階層的プランニングの手法に基づいてText Planningを行っている。プランニングの結果として、ゴールを頂点とし各葉が個々の文に対応する、「プランの木構造」が得られる。この木構造のレベルの深さが説明の詳細さを表している。

またプラン修正部(critics)は、ユーザの理解度に応じて説明の詳細さを変えるために、ユーザモデルを参照して、既知概念について冗長な説明を行わないようにプランを修正する。プラン生成部で得られる「プランの木構造」は説明の詳細さ

を表しているため、ある概念についての説明を省略する処理は、対応する部分木を取り除くことで実現できる。

以上のようにして得られる「プランの木構造」から、つながりのある読みやすい文章を生成するために、プラン洗練部は説明文中の焦点の流れを管理している。プラン洗練部は、焦点の流れから代名詞による照応指示や主語化等の表層構造の決定を行い、文章のrefinementを行う。

3 知識表現形式

本システムでは、知識表現形式としてフレーム表現を用いている。概念間の関係を表す概念構造として、概念フレームおよび文フレームと呼ぶ2種類のフレームを用意している。概念フレームは、概念間の静的な階層関係を表現したもので、概念の定義を与えるものとみなすことができる。文フレームは、概念フレームで表現できない概念間の関係を格関係に基づいて表現したもので、文の意味表現をなしている。文フレーム中に現れる対象概念は、概念フレームからのポイントによってつながれており、このポイントを用いて特定の概念フレームから関連する文フレームを取り出すことができる。図2に「アセンブリ言語」と「機械語」の概念フレーム、およびそれらに関連する文フレームの例を示す。

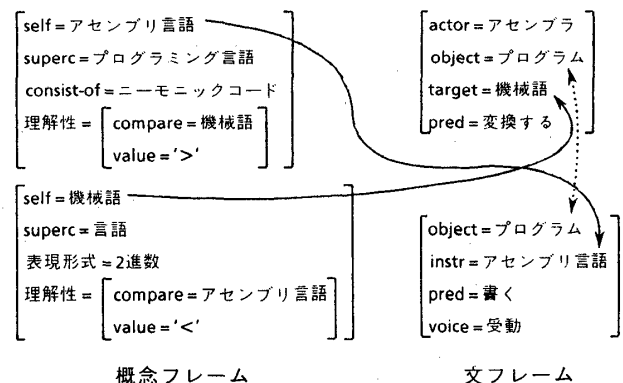


図2 概念フレームと文フレーム

4 プラン生成部

4.1 Text Planning

ユーザからの質問の型として、本システムでは以下の5種類の質問を取り扱っている。

- What-is型 ... 「アセンブリ言語とは何ですか?」
- Difference型 ... 「アセンブリ言語と機械語の違いは?」
- How型 ... 「ファイルを消去するにはどうすればよいか?」
- Similarity型 ... 「アセンブリ言語と機械語の共通点は?」
- Relation型 ... 「アセンブリ言語と機械語の関係は?」

このうち本稿では、特にWhat-is型、How型、Difference型の質問についてのみ考察する。これらの質問形式に応じてそれぞれ以下のゴールが得られる。

DEFINE型ゴール

What-is型の質問に現れる概念に定義を与える

TO-DO型ゴール

How型の質問に含まれる作業を達成する手順を与える

DISTINGUISH型ゴール

Difference型の質問に現れる2つの概念間の区別を行う

たとえば、質問「アセンブリ言語とは何ですか?」からは、ゴール'DEFINE(アセンブリ言語)'が得られる。また、質問「ファイルを消去するにはどうすればよいか?」からは、ゴール'TO-DO(消去(ファイル))'が得られる。

このようなゴールは、いくつかの作業を組み合わせることにより達成することができる。ゴールを達成するための作業をプランと呼ぶ。また、ゴールとプランの組み合わせをオペレータと呼ぶ(図3)。ゴールからプランへの展開は、与えられたゴールとマッチするオペレータを取り出し、ゴールを根とする「プランの木構造」を展開する形で行われる。

プランは、システムが直接実行可能なプリミティブなもの、他のオペレータによってさらに展開されるものに分けられる。「プランの木構造」の展開は、長さ優先(length-first)で行われ、すべての葉がプリミティブなプランOUTPUTに展開された時点で終了する。このプリミティブなプランを左の葉から順に並べたものが最終的に得られるプラン列である。

4.2 DEFINE型ゴールからのプランニング

DEFINE型ゴールは、概念構造から説明の対象となっている概念の属性情報を取り出して、表層構造として出力するプラン列に展開される。DEFINE型のゴールに適合するオペレータは、図1中のオペレータ1である。オペレータ1は、

「Xという概念を定義するには、その上位概念と構成要素について説明し、他の属性があるときはそれらの属性内

オペレータ1	GOAL: DEFINE(X) PLAN: EXPLAIN(X){superc=A} (-1), EXPLAIN(X){consist-of=B} (-1), Xをその他の属性により説明する (-1), Xをその文フレームSにより説明する (-1)
オペレータ2	GOAL: TO-DO(X) PLAN: Xと適合するオペレータYを検索する (0), OUTPUT(Y) (0), TO-DO(Y)に含まれるプラン (-1)
オペレータ3	GOAL: DISTINGUISH(A,B) PLAN: DEFINE(A) (VA), DEFINE(B) (VB), AとBの間の概念的な関係を説明する (0)
オペレータ4	GOAL: EXPLAIN(X){R=Y} PLAN: OUTPUT(X){R=Y} (0), DEFINE(Y) (VY)
オペレータ5	GOAL: 消去(ファイル) PLAN: DO-COMMAND(remove(ファイル)) (0)

図3 プラン展開用オペレータ(一部分)

容についても述べ、さらにXについての文フレームがあればその文フレームの内容を述べればよい。」

ということの意味している。

属性について説明を行うEXPLAIN型のプランはオペレータ4と適合する。オペレータ4は、

「概念Xをその属性RがYであることにより説明するには、属性Rについての説明文を生成したあと、新たな概念Yに話題を移して説明を行えばよい。」

ということの意味している。

4.3 TO-DO型ゴールからのプランニング

TO-DO型ゴールは、引数として与えられた作業を達成する手段を説明するプラン列に展開される。TO-DO型ゴールに適合するオペレータは、図1中のオペレータ2である。オペレータ2は、

「作業Xを実現する方法を示すには、XをゴールとするオペレータYにその方法が示されているので、Yの内容について述べたあと、Yに含まれるサブゴールを実現する方法を示せばよい。」

ということを示している。たとえば、ゴール'TO-DO(消去(ファイル))'がオペレータ2と適合した結果、Xは'消去(ファイル)'で束縛され、Yは'消去(ファイル)'をゴールに持つオペレータ5で束縛される。この結果、オペレータ5自体を表層構造として出力するプランと、オペレータ5のプランを引数とするTO-DO型のプラン'TO-DO(DO-COMMAND(remove(ファイル)))'に展開される。

4.4 DISTINGUISH型ゴールからのプランニング

DISTINGUISH型ゴールは、引数に与えられた2つの概念について定義を行うプランと、2つの概念間の関係を推論するプランに展開される。ゴール'DISTINGUISH(A,B)'が与えられたときは、オペレータ3にみられるように、まずAとBについてDEFINE型のプランによる定義を与える。その後、連想メカニズムによってAB間の関係をあらわすリンクをたどり、各概念間の関係を推論する。連想メカニズムによって推論された関係は、最終的にはプリミティブなプランOUTPUTによって表層化される。

5 まとめ

本稿では、プランニングの手法を用いてユーザの理解度に応じた説明文を生成する、マニュアルレスシステムについて述べた。本システムは、ユーザとの対話を介してユーザモデルを構築し、プランニングにフィードバックさせることによって、説明の詳細さを変えるようになっている。

現在、質問からゴールを推論する処理が充実していないために、質問の型が限定されている。他の質問形式を取り扱うためには、質問文からユーザの意図を抽出するという処理が必要である。さらに、プランニングで用いるオペレータを拡充しなければならない。

参考文献

- [1] McKeown, K. R.: *Text Generation*, Cambridge University Press (1985).
- [2] Wilensky, R. et al.: *Talking to UNIX in English: An Overview of UC, Comm. of ACM, Vol.27, No.6, pp.574-593 (1984).*