

# UNIX日本語インタフェースにおける 履歴情報の利用

西島 恵介<sup>†</sup> 矢野 敏之<sup>†</sup> 久保 康司<sup>‡</sup> 藤田 米春<sup>†</sup>

<sup>†</sup> 大分大学 工学部 <sup>‡</sup> 山口大学大学院

本報告では、UNIX日本語インタフェースにおいて、ユーザの要求文の解釈に履歴情報を利用する方法について検討する。

これまでに、ユーザから入力される要求文のうち、主に単文に関してインプリメントを行なっている。しかし、ユーザがより自然にコンピュータと対話を行なうには、単文だけでは不十分である。今回、複文、重文への拡張を行い、語の省略、照応、不完全な文の処理について検討を行なった。また、この処理を入力系列の履歴に対して使用し、次の入力の予測等を行なう手法について検討する。

## Use of History Information in Japanese Language Interface for UNIX

Keisuke Nishijima<sup>†</sup> Toshiyuki Yano<sup>†</sup> Kouji Kubo<sup>‡</sup> Yoneharu Fujita<sup>†</sup>

<sup>†</sup> Faculty of Engineering, Oita University

<sup>‡</sup> Graduate School of Engineering, Yamaguchi University

In this paper, use of history information in Japanese language interface for UNIX is described. History information is used for interpretation of input sentences.

A Japanese language interface which interprets simple sentences is already implemented. However, only simple sentences are not enough to talk with computer naturally. Therefore, the Japanese language interface is extended to interpret complex and/or compound sentences. In addition, handling of ellipses, anaphoras and incomplete sentences is presented. Furthermore, use of history information for forecast of next input sentence is studied.

## 1. はじめに

我々は、エンドユーザの負担を軽減するために、知的アシスト機構を持った日本語インターフェースの構築を目指している。現在、UNIXシステムにおけるファイル操作を対象としたインターフェースシステムを構築中である。[1]-[5]

これまでに、ユーザから入力される要求文のうち、単文についてのインプリメンツを行っている。しかし単文だけではユーザが、より自然にコンピュータと対話を行なうには不十分である。今回、複文・重文への拡張を行った。それに伴い、語の省略、照応への対応、不完全な文の処理、要求文が連続して入力された時の履歴情報の利用について検討を行ったので、報告する。

## 2. システムの構成

システムの構成を図2.1に示す。以下に、それぞれの構成部の機能、そこで用いられる知識について述べる。

### ユーザ対話部

ユーザ対話部はユーザとの入出力のやり取りを行なう。ユーザからの入力文は要求文解析部に渡される。また、ユーザへの問い合わせ、応答が必要な時は、適切な文を生成し、対話を行なう。

現在、ユーザ対話部についてはインプリメンツされていない。

### 要求文解析部

まず、名詞・動詞辞書等により語彙解析が行われ、入力文を単語へ分割する。その時、品詞情報も付加しておく。次に、意味マーカ辞書等により構文・意味解析を行い、文の表層的な格構造表現を生成する。生成された意味表現は意味意図解析部へ渡される。また、入力文が複文・重文の場合、文の分割や制約条件の付加を行なう。

### 意味・意図解析部

領域知識、履歴情報を用いて、対象領域における意味、ユーザの意図を表わす意味構造表現に変換する。

ここでは、履歴情報をを利用して、省略語の補完、代名詞の照応、不完全な文の補完処理も行われる。

### コマンド生成部

得られた意味構造表現をもとに、コマンド知識、生成プラン、コマンド結合戦略を用いてコマンド系列を生成する。

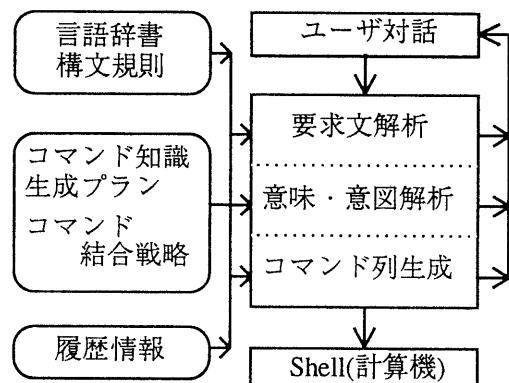


図2.1 システム構成

以下、本報告では、要求文解析部における重文・複文処理への拡張、意味・意図解析部における履歴情報の利用について詳しく述べる。

## 3. 要求文解析部

要求文解析の目的は、コマンド生成に必要な文法情報を抽出し、文の表層的な格構造表現を生成することである。以下に単文の解析、複文・重文の解析への拡張について述べる。

### 3.1 単文の解析

### 3.1.1 単文の分析

これまでにユーザから入力される要求文のうち、主に単文を対象に分析を行っている。その分析結果を簡単に述べる。

#### 係り受け

文の長さは比較的短い。したがって、係り受けは比較的単純である。

#### ユーザの意図

ユーザの発話意図により、要求文は3つのタイプに分類できる。この分類は要求文解析部において、生成する格構造の要素に「様相格」として付加する。

質問: 「～できますか？, ～ですか？, …」  
要求: 「～して下さい, ～しなさい, …」  
説明: 「～です, はい, いいえ, …」

以下、主に「要求」の文について分析を行なっている。

#### 意味マーカ

名詞の意味や属性を分類し、「場所」、「操作対象」、「属性」等の意味マーカを定義した。また、意味マーカ同士の関係を上位一下位、部分一全体関係で階層化している。これらは、名詞の係り受けの決定や動詞の格構造と名詞句の関係を決定するための制約の1つとして用いられる。

#### 動作格

動詞の動作を分析することにより、以下に示す2種類の動詞に分類した。この分類は動詞の格要素に「動作」格として付加し、複文処理に用いられる。

操作: 対象の操作 「削除する, 移動する, …」  
条件: 対象の制約 「含む, 持つ, …」

### 3.1.2 単文の処理

まず、単文は語彙解析により品詞情報を決定した後、句構造文法により名詞句の係り受けが決定される。係り受けの決定には以下の規則を用いる。

- (1)意味マーカの組み合わせによる制約  
条件に合うものを優先する
- (2)語句同士の距離が近いものを優先する
- (3)属性は重複して定義されない
- (4)係り受けは交差しない

次に、動詞の格構造に基づき、動詞句と名詞句の関係を決定することにより格構造表現が得られる。ただし、動詞の格には必ず値を持たなければならない必須格を定めている。例えば、「コピーする」という動詞は、対象格と場所格が必須格である。値が決まらない時は、補完処理により決定する。以下に処理例を示す。

入力	ディレクトリ b のファイル c を ファイル d にコピーしろ
解析結果	
動詞:	コピーする
対象:	[ファイル, [名前, c ], [場所,[ディレクトリ,[名前, b ]]]]
場所:	[ファイル,[名前, d ]]
動作:	操作
様相:	要求

以上の結果をもとに、重文・複文処理への拡張を行った。以下に詳しく述べる。

### 3.2 要求文解析部の拡張

#### 3.2.1 複文処理

複文の場合、主語・述語関係が2回以上成立し、従属節が連体修飾節になる。

例) bで始まるファイルを表示しろ

この例では「bで始まる」という従属節が「ファイル」を修飾している。これは、「ファイル」の「名前」という属性に「bで始まる」という制約を加えていると考える事ができる。

したがって、「bで始まる」を「ファイル」の「名前」に制約として加えることにより、1つの格構造で表現できる。この場合、「名前」という単語が文に現れていないが、ファイルの属性のデフォルト値を参照することにより付加できる。(4.1参照のこと)

例) ソートしたファイルを表示しろ

この例でも同様に「ソートした」という従属節が「ファイル」を修飾しているが、この文は過去に「ファイルをソートした」事実があり、「そのファイルを表示する」と解釈できる。

この例では、「ファイルをソートした」、「そのファイルを表示する」の2つの単文の系列として処理し、1つ目の文の動詞が過去形であるという情報を利用することにより、履歴情報を検索し、対応するファイルを表示することができる。

この2つの例の違いは、従属節に含まれる動詞の機能が対象を制約する動詞であるか、対象を操作する動詞であるのか、にある。これらのことから、複文は従属節の動詞の動作格の種類により、以下のような処理を行なう。

条件：修飾節として主節の制約として処理し、1つの格構造として生成する。  
操作：2つの文として処理し、2つの格構造として生成する。

2つの文として処理する場合、省略語の補完、代名詞の照応処理等を行なう必要があるが、これについては4.で述べる。

### 3.2.2 重文処理

重文の場合、主語・述語関係が2回以上成立し、それらが対立節となる。

例) ファイルbをディレクトリcにコピーして、ディレクトリcの内容を表示しろ

この例では、「ファイルbをディレクトリcにコピーする」、「ディレクトリcの内容を表示する」という2つの単文の連続と解釈できる。

例) ファイルbをソートして、ファイルcに保存しろ

この例も「ファイルbをソートする」、「その結果をファイルcに保存する」という2つの単文の連続と解釈でき、前方の単文の実行結果を次の単文の処理対象することにより処理できる。

この例では「ファイルcに保存しろ」と保存する対象が省略されているが、この場合は「ファイルbをソートした結果」ということになる。

以上から、重文は単文に分割し、単文の連続系列として処理できることがわかる。

しかし、処理対象や場所等が省略されたり、代名詞で置き換えられることが多い。

このため、省略語の補完や代名詞の照応処理等を行なう必要がある。これらについては、4.で詳述する。

### 3.2.3 不完全な文の処理

要求文解析部では、動詞の格構造に基づき処理を行なうが、動詞が省略されたよう

な要求文も考えられる。

このような場合、要求文解析部では動詞の含まれない構造を生成し、意味・意図解析部において履歴情報を参照し、動詞の補完処理を行なう。（4.3参照のこと）

また、動詞は存在しないが、文末の名詞が動作をあらわす名詞の時は、動詞とみなして処理を行なうことにより不完全な文の処理を行なう。

#### 4. 意味・意図解析部

意味・意図解析部の目的は、要求文解析部で得られた格構造表現をもとに、省略語の補完や代名詞の照応処理等を行い、コマンド生成に必要な意味構造表現を生成することである。

以下に省略語の補完、代名詞の照応、不完全な文の補完、次の入力の予測処理について述べる。

##### 4.1 省略語の補完処理

3.で述べたように、複文・重文では後続の節で、語の省略が起こる。また、連続して入力される要求文においても語の省略が起こる。

本システムはユーザの要求を実行するため、実行する対象や場所等の情報が必要である。これらは、入力された文の動詞の必須格を参照することにより、何が必要か判断できる。

例) ファイル b をディレクトリ c に  
コピーしろ。

ディレクトリ d にコピーしろ。

この例では、まず、ファイル b の存在場所が省略されている。UNIXでは異なるディレクトリに、同じ名前のファイルが存在することがあるので、補完する必要がある。

もし、前方に補完する候補がないときは、

システム内のファイルの知識のデフォルト値を参照し、補完する。本システムでは、ファイルの存在場所のデフォルト値はカレントディレクトリである。

また、2つめの文の対象格が省略されている。これは、「コピーする」という動詞の必須格であるから補完する必要がある。

したがって、動詞の必須格が埋まっているかをチェックし、埋まっていない格に対しては補完を行なう。以上のことから、省略語の補完は以下の規則により行なう。

- (1) 前方の文に補完可能な格要素が見つかれば、その内容により補完する。
- (2) 補完される語は、補完する格要素の意味マーカと適合しなければならない。
- (3) 係り受けを行なう語が存在する場合、係り受けのチェックを行なう。
- (4) 補完する候補が複数ある場合、文の距離が近いものを補完する。
- (5) 補完すべき格にデフォルト値が存在する時は、それを補完する。

規則を上から順に適用し、省略語の補完を行なう。

この例では、「ファイル b」が補完の候補となり、条件を満たすので、2つめの文は「ファイル b をディレクトリ d にコピーしろ」となる。

また、この規則で補完できないものは、ユーザに問い合わせを行い、補完する。

##### 4.2 照応処理

代名詞の照応についても、省略語の補完処理と同様に行なう。代名詞に続く助詞により、どの格の照応を行なえばよいかが決定できる。照応の候補が見つかると、意味マーカと係り受けのチェックを行ない、条件に合えば照応処理を行なう。

以上のことから、代名詞の照応処理は以下の規則により行なう。

- (1)代名詞に続く助詞により照応すべき格を決定し、前方の文に照応が可能な格要素が存在すれば、その内容と照応する。
- (2)照応される語は、動詞の照応する格の意味マーカと適合しなければならない。
- (3)係り受けを行なう語が存在する場合、係り受けのチェックも行なう。
- (4)照応する候補が複数ある場合、文の距離が近いものと照応する。

例) ファイル b を表示しろ。  
それを削除しろ。

この例では、助詞「を」から対象格の内容を参照すればよいことがわかる。動詞「削除する」の対象格の意味マーカとファイル b はマッチするので、2つめの文は「ファイル b を削除しろ」となる。

#### 4.3 不完全な文の補完処理

4.1の省略語の補完では、動詞をキーにして、補完を行なっていた。しかし、動詞が省略されたような要求文も考えられる。

例) ファイル b をディレクトリ c にコピーしろ。  
ファイル c も。

このような例では、「ファイル c も」という文から「ファイル c について前の処理を繰り返す」と解釈できる。このような場合、「ファイル c も」からファイル c が対象格に対応することがわかるので、前方の文の対応する格の内容を入れ換えて実行することにより補完できる。

したがって、不完全な文の補完処理は以下の規則により行なう。

- (1)得られる格要素について前方の文の同じ格の内容と置換を行い、文を生成する。
- (2)置換する内容はその格の意味マーカと適合しなければならない。
- (3)係り受けを行なう語が存在する場合、係り受けのチェックも行なう。
- (4)照応する候補が複数ある場合、文の距離が近いものと照応する。

この補完処理は不足している情報を履歴情報により補うものであり、必要な情報のすべてが履歴から得られるわけではないので、必ずしもユーザの意図した処理を実行できるとは限らない。

このような場合、実行前にユーザに問い合わせを行なう必要がある。もしくは、実行の取り消しを行なえる機能が必要であるが、本システムでは未考察の機能である。

例) ファイル b をファイル c にコピー

また、上記の例のように、動詞は存在しないが、文末の名詞が動作を表わす名詞の時は、要求文解析部において、動詞とみなすことにより処理を行う。

#### 4.4 次の入力の予測処理

入力された要求文の履歴を利用することにより、次の様なことが考えられる。

- (1)コマンド生成をせずにコマンド列を決定する。
- (2)次に入力される要求を予測することにより、コマンドの検索範囲の限定を行なう。

(1)については、意味構造表現とコマンド列の組みを履歴情報として残すことにより実現できる。

この場合、意味構造表現のマッチングを行なう必要があるが、格要素と意味マーカ

とによるマッチングにより行なう。ここで、履歴情報として残す時に、抽象化を行っておけば、比較的簡便に処理できると思われるが、今後の検討が必要である。

(2)についても同様であるが、履歴とマッチングを取ることにより、どのような作業の途中であるかが把握できると考えられる。マッチした履歴の次の入力を予測候補とすることにより、検索範囲の限定を行なえると考える。しかし、どのくらいの履歴を残しておく必要があるのか、似たような履歴はどのように抽象化するのか、といった問題がある。<sup>[6]</sup>

この場合、対話の文脈を理解する必要があり、履歴のどの部分がどういう作業を目的とした一連の作業であるか、ユーザの話題の焦点や意図がどのように変化しているかを把握する必要がある。意味・意図解析部は現在のところ、意図の把握のレベルには達しておらず、今後の検討が必要である。

## 6. おわりに

今回、単文の処理から複文・重文や不完全な文の処理へを拡張を行い、語の省略や照応等に対応し、ユーザが以前よりは、より自然にコンピュータと対話を行なえるようになった。現在、複文・重文処理は実装済みであり、不完全な文の処理、次の入力の予測処理は現在、実装中である。

本システムはPrologを中心にC言語を併用しインプリメントを行なっている。また、現状では要求文はキーボードを用いて入力しているが、ユーザの入力の簡便さを考えると音声入力を用いる必要がある。

現在、ユーザの意図を把握するメカニズムは、検討中の段階であり、より一層の検討が必要である。

次の入力の予測処理については、複数のシステムを扱うインターフェースでは必ず必要となると考えられる。

本システムは、現状では、UNIXのファイル操作のみを扱っている。現在、複数システムに対しての拡張を検討中である。

## 謝辞

本研究に協力して頂いた神山文子技官、長野信二、岩見貴弘の各氏に感謝致します。なお、本研究の一部はトーメン情報システムズ(株)との共同研究「知的ヘルプ機能を持った、OSの自然言語インターフェースの研究」によるものである。

## 参考文献

- [1]渡辺他:UNIX日本語インターフェースシステムの構築,平成3年電閣九支連大, No.1214(1991)
- [2]西島他:UNIX日本語インターフェースシステムの構築-要求文からの意味構造の抽出-,平成4年電閣九支連大,No.1208 (1992)
- [3]久保他:UNIX日本語インターフェースシステムの構築-コマンド辞書の記述とコマンドの選択-,平成4年電閣九支連大, No.1207(1992)
- [4]矢野他:UNIX日本語インターフェースシステムの構築-意味構造表現とコマンド結合-,情報処理学会第47回全国大会論文集,第5分冊,pp287-288(1993)
- [5]久保他:語彙の領域固有の機能に基づく日本語要求文の解釈,情報処理学会九州支部研究会,pp.1-6(1993)
- [6]久保他:コマンド連鎖グラフを用いたコマンド系列の予測,平成5年電閣中支連大,p.355(1993)
- [7]R.Wilensky et al.:  
The Berkeley UNIX Consultant Project,  
Computational Linguistics, Vol.14, No.4,  
pp.35-84(1988)
- [8]田中:自然言語解析の基礎,産業図書,(1989)

## 付録 システムの実行例

```
[15:53]csis-c:~/Bin> Jsh  
Loading file .... wait a moment  
....Done
```

```
Japanaese Interface for UNIX System  
Version 1.0 December 1993  
start.....
```

入力 > pで始まるファイルを表示して下さい

格構造 > [[表示する,動詞,操作], [対象, [ファイル, [条件, [始まる, [値,p]]]]]]]

意味構造表現>[out([attr\_all,[file,[name,begin(p)]],[place,'.']]],stdout)]

生成コマンド > 'ls -il ./ | awk '\$9 ~ /^p/ {print}''

```
739598 -rwx----- 1 nisijima      53 Feb  9  1993 pog  
739597 -rwx----- 1 nisijima      50 Feb  9  1993 pom  
739649 -rwx----- 1 nisijima     616 Aug 12 15:10 priset  
746439 drwx----- 2 nisijima      512 Jul  1  1992 ps  
739702 -rwx----- 1 nisijima    24576 Dec 18  1992 pspages
```

> コマンドの実行を終了しました。

入力 >