

## アプリケーションの状態に依存したガイドを提供する汎用モジュール

森 俊也 北野 幹浩 経田 樹朗

松下電器産業(株) 情報システム研究所

近年、ワープロ、パソコンをはじめとする情報機器が非常な勢いで普及してきている。かつては、一部の人がだけ使用していたものが、現在では家庭にまで入り込んでいる。それに伴い、様々なユーザに対応すべく、様々な機能を有したアプリケーションが開発されている。この高機能化、多機能化に伴い、非常に便利になったと考えられがちであるが、操作方法が複雑になり、かえって使いづらいものになっている。我々は、現在のユーザインタフェースの抱える問題点(1. コマンド検索が柔軟にできない。2. 状況に応じたガイドがない。3. ユーザの習熟を促す機能がない。)を明らかにし、この問題を解決する方法として、コマンド検索が柔軟にでき、かつ状況に応じたガイドを提示するパッシブガイド、ユーザにコマンドを積極的に提示するアクティブガイドの開発を行なった。

**A Module providing Context Sensitive Guidance  
with Application Program**

Toshiya Mori, Motohiro Kitano and Tatsuro Kyoden

Information Systems Research Laboratory,  
Matsushita Electric Industrial Co., Ltd.  
1006 Kadoma, Kadoma City, OSAKA, JAPAN, 571

These days word processors and personal computers are coming into wide usage. They were used by special people in the past, now they come into general usage. As a result, they have various functions and tools for various users. They are becoming high-functional and complex. In this report, we made it clear that computer systems had major problems (e.g. difficulty in command retrieval, canned online help messages and no function to elevate the user's knowledge about application) in user interface. So we developed active guidance and passive guidance. Passive guidance is to produce command retrieval and context sensitive help. Active guidance is to offer the most helpful command to users.

# 1 まえがき

現在のアプリケーションは、高機能化、多機能化に伴い、操作方法が複雑になってきている。一方、アプリケーションは、ユーザの習熟を支援する機能を十分備えているとはいえず、その為に、一般ユーザは、マニュアルを熟読しないとアプリケーションを使いこなすことができない。また、熟練者であっても、一部の高性能しか使っていないのが現状である。

我々は、マニュアルを熟読せずに使えるアプリケーションを容易に開発する為に、アプリケーションに知的なガイド機能を提供する汎用モジュールの開発を行なった [経田 91] [Kyoden 92]。本稿では、この汎用モジュールの内容について報告する。

従来のアプリケーションに実装されているヘルプ機能は、ユーザがコマンド名で検索することにより行なっている。これは、マニュアルの延長上であり、どのコマンドを用いて良いのか分からないユーザにとっては、何処を調べてよいのかが分からない。また、現在のように高機能化、多機能化が進んでくると、マニュアルは数十冊にも及び、それらのマニュアルにすべて目を通してから、操作するというのでは、大変ばかりである。また、分からないコマンド等が出てくると、その度に、マニュアルを調べないといけないのでは、せっかくの高機能化、多機能化が逆に使いづらいものになっていると考えられる。

近年、より使いやすいヘルプ機能が開発されるようになってきたが [吉良 88][宮崎 89][斉藤 87][鈴木 88]、現状のアプリケーションのヘルプ機能は、コマンド名またはメニューで検索する方法であり、どれもマニュアルの域を脱していない。また、アプリケーションの状態を考慮してコマンド名で検索するものも出てきたが、どれもまだユーザにとって使いやすいものとは言えない。そこで我々は、以下の機能を有する知的ヘルプシステムの開発を行なった。

- 自然言語入力  
自然言語でユーザが自由に質問入力できる。但し、アプリケーションのコマンド実行を問い合わせる簡単な文に対してのみである。
- パッシブガイド機能  
ユーザの操作履歴からユーザの習熟度を推定し、アプリケーションの状態を把握することにより、ユーザの要求に対してユーザのレベル、

アプリケーションの状態に依存したガイドを提示する。

- アクティブガイド機能  
ユーザの操作履歴から、非効率な操作を行なった場合、効率的な操作を積極的に提示する。これによりユーザが今まで未知であった操作に対して、学習することが可能となる。
- 汎用性  
各アプリケーションに合った知識ベースを用意することにより、アプリケーションに独立で、かつ汎用モジュールとして実現可能である。

# 2 モジュールの構成

本モジュールの構成は、ユーザとアプリケーションとの間に介在するのではなく、アプリケーションからモジュールが呼び出される形態をとっている。このモジュールはアプリケーションの1つのライブラリとして存在している。図1にモジュールの構成図を示す。

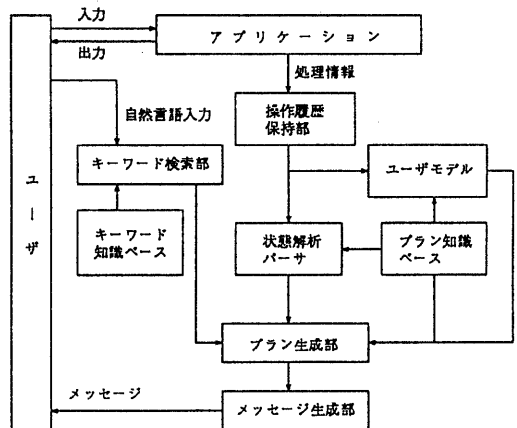


図1. モジュールの構成図

アプリケーションから処理情報が送られ、その情報は一旦操作履歴に保持される。その操作履歴から状態解析パーサがプラン知識ベースのパーサを行ない、アプリケーションの状態とプラン知識ベースとのマッピングを行なう。このプラン知識ベースとは、CFG (Context Free Grammar) で表現された知識ベースであり、詳細は後述する。

プラン知識ベース中の各ノードにはユーザモデルがあり、状態解析パーサがパースすることにより、ユーザモデルの値（既知、未知）の設定を行なう。

パッシブガイドはキーワード検索部、プラン生成部、メッセージ生成部によって処理される。キーワード検索部は、ユーザが入力した質問文からプラン知識ベースのノードを確定する。ここでの質問入力の内容は、ユーザが実行したい操作のみについてである。次にプラン生成部は、現在のパース結果からユーザの要求するノードまでのパスを調べる。そのパスを元に、メッセージ生成部がメッセージの生成を行なう。詳細については後述する。

アクティブガイドは、状態解析パーサのパース結果から、ユーザが非効率な操作を行っていた場合、メッセージ生成部がメッセージの生成を行なう。詳細については後述する。

本モジュールは、ワークステーション上の日本語ワープロにインプリメントを行なった。モジュールのプログラムサイズは180KBであり、プラン知識ベースは400ルールで約200KBである。以下、通信方法、プラン知識ベース、状態解析パーサとユーザモデルについて詳細に述べる。

## 2.1 通信方法

アプリケーションの状態に依存したガイドを提供する為には、アプリケーションの状態を把握する必要がある。ここでは、本モジュールの通信方法について説明する。この通信方法は、図2に示すようにイベント抽出とプラン変換に分かれる。ユーザの入力に対して、アプリケーションがイベントを処理するところに処理情報を送るよう設定する。その処理情報は、処理情報テーブルにより後述のプラン知識ベースの終端記号と変数に変換される。この処理情報テーブルとは、アプリケーションから送られてくる処理情報と、プラン知識ベース中の終端記号と変数との関係を表したテーブルである。変換された終端記号と変数は状態解析パーサに渡され処理される。

## 2.2 プラン知識ベース

アプリケーションのコマンドを複数組み合わせでプランとして記述し、そのプランをCFGで表現した知識ベースである。このプランは終端記号、非終端記号、条件文、アクティブガイド用のメッセー

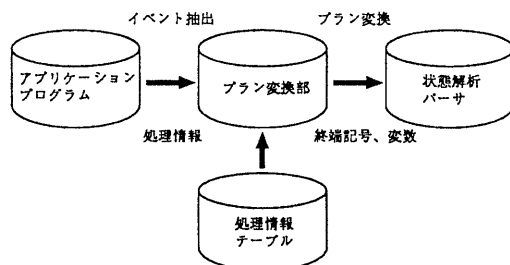


図2. アプリケーション - モジュール間通信

ジで表され、BNF (Backus Naur Form) で記述されている。終端記号は、終端記号名と変数、非終端記号は、非終端記号名と変数で表されている。終端記号名は、ユーザの入力（ファンクションキー、メニュー選択、カーソル移動等）に相当し、非終端記号名は、ユーザの入力（終端記号）によって実行されるプランに相当する。また、変数はプランに必要なアプリケーションの処理情報を表す。条件文は、前述の変数で表された論理式として記述される。図3にプラン知識ベースの文法仕様を示す。図中の\*は、0回以上の繰り返しを意味する。なお条件文が無い場合、変数は省略される場合がある。

```

<プラン> ::=
  <左辺> -> <右辺> | <右辺> *
<左辺> ::=
  <<非終端記号名> <変数>* >
<右辺> ::=
  [ <<非終端記号名> <変数>* > | ( <終端記号名> <変数>* ) *
  [ [ [ <条件文> ] [ <ガイドメッセージ> ] ]
  [ [ <ガイドメッセージ> ] ]
  [ ( ) ]
  
```

図3. プラン知識ベースの文法仕様

このプランには、パッシブガイド用のプランとアクティブガイド用のプランがある。パッシブガイド用のプランは、終端記号または非終端記号で構成され、ユーザに対してガイドを提示するプランを表す。またアクティブガイド用のプランは、終端記号、非終端記号、条件文、ガイドメッセージで構成され、効率の良いプランと非効率なプランに分けられる [Desmarais 87]。プラン知識ベースの具体例を図4に示す。

ここで挙げたのは、日本語ワープロの典型的な例である。パッシブガイド用のルールは文書を作成するのに、まず文書入力を行ってから編集をし、

```

<S>-><文書作成>
<文書作成>-><文書入力> <編集> <ファイル保存> <印刷> (終了)
<文書入力>->(文字入力)
-><ファイル入力>
<編集>    -><編集>(削除)
          -><編集>(移動)
          -><編集>(複写)
          ->(削除)
          ->(移動)
          ->(複写)
<ファイル保存>->(保存メニューの選択) <ファイル名の指定>
<印刷>    ->(印刷メニューの選択)
<ファイル入力>->(呼び出しメニューの選択)
<ファイル入力>->(呼び出しメニューの選択) <ファイル名の指定>
<ファイル名の指定>->(ファイル名入力)(実行キー)
(注) <S> : 開始記号
    
```

```

<移動>-><文字列移動>
-><削除><貼り付け>
("削除した文字を貼り付けるには
"文字列移動"を用いる方が便利です。")
    
```

図4. ブラン知識ベースの具体例

ファイル保存、印刷を実行してから終了することを表している。また、アクティブガイド用のルールは文字を移動する場合で、文字列移動コマンドを使用する効率の良いプランと、削除コマンドと貼り付けコマンドを使用する非効率なプランで表されている。

### 2.3 状態解析パーサとユーザモデル

アプリケーションの状態を把握し、かつユーザに対して効果的なガイドを提示する為には、アプリケーションの状態とブラン知識ベースとのマッピングを行なう必要がある。ここでは、アプリケーションから送られてくる終端記号名と変数からブラン知識ベースのパーサを行なう状態解析パーサと、そのパーサ結果によってユーザの習熟度を推定するユーザモデルについて説明する。

この状態解析パーサはチャート法によって実現されている。このチャート法は逐次型で、入力である終端記号名と変数を順に読み込みながら活性弧を生成し統語解析を行なう [Kay 80]。また本状態解析パーサは、富田法 [Tomita 86,87] に見られるようなスタックの統合によるメモリーの圧縮や、シフト

操作よりリデュース操作を優先するを行なっている。またリデュース操作を効果的に行なう為に、到達可能行列 [Aho 72] [Platt 73] を用いている。図4のパッシブガイドのルールに対して入力:"文字入力","複写"をパースしたときの結果を図5に示す。

入力: 0 文字入力 1 複写 2

```

0 1 <文書作成> -> <文書入力>・<編集> <印刷> (1)
      <ファイル保存> (終了)
1 2 <編集>    -> <編集>・(削除) (2)
1 2 <編集>    -> <編集>・(移動) (2)
1 2 <編集>    -> <編集>・(複写) (2)
0 2 <文書作成> -> <文書入力> <編集>・<ファイル保存> (2)
      <印刷> (終了)
    
```

図5. パース結果

入力:"文字入力","複写"に対して順にパースを行なう。"文字入力"に対してパースを行なった結果が(1)であり,"複写"に対してパースを行なった結果が(2)である。ここでの"."の左項はすでにパースされたことを示している。

次にユーザモデルについて説明する。このユーザモデルはブラン知識ベースにユーザの習熟度を反映させるもので、ブランの右辺の終端記号、非終端記号にユーザモデルを付加したものである。このユーザモデルとは、既知、未知の値を持ち、回数とパース時間によって設定される [Green 85][永田 87]。この回数とは、前述のパーサによってパースされた終端、非終端のパース回数であり、1回パースされるごとに1増加される。なお再帰ルールによる同一記号の複数のパースは、1回とみなすように設定している。パース時間とはノードがパースされた時間を示している。ユーザモデルの初期値はすべて未知で設定されており、回数とパース時間によるユーザモデルの設定は、以下の図6、図7によって行なう。

図6は、回数によるユーザモデルの設定方法である。回数がN2以上になると既知に設定され、回数がN1以下に減少すると未知に設定される。

図7は、経過時間Tによる回数の減少方法である。現在パースされた時間が、以前パースされた時間からT時間経過していると、NはN/2に減少する。もし2T時間経過していると、NはN/4に減少する。回数Nは経過時間Tの倍数に反比例して減少する。

またユーザの質問から、ある終端または非終端が明らかに未知と分かった場合、その終端または非終端の回数はN1に、ユーザモデルは未知に設定される。

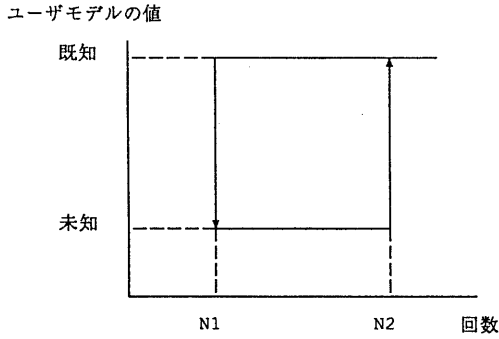


図6. 回数によるユーザモデルの設定方法

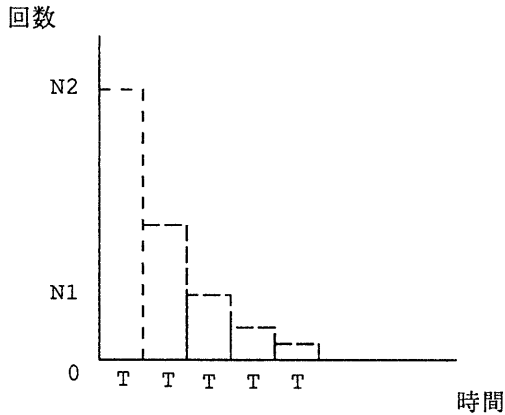


図7. 経過時間Tによる回数の減少方法

### 3 パッシブガイド

パッシブガイドは、ユーザの自然言語による質問文からユーザの要求するノード（プラン知識ベース中の終端記号、非終端記号）を抽出し、状態解析パーサの現在のパス結果からそのノードを実行するまでのパスを調べ、パスの結果からメッセージの生成を行う。処理は大きくキーワード検索、プラン

生成、メッセージ生成に分かれる。以下これらについて順に説明する。

#### 3.1 キーワード検索部

ユーザが自然言語で入力した質問文からキーワードを抽出し、そのキーワードがどの代表語に含まれるかを調べ、その代表語からプラン知識ベース中のノードを確定し、ユーザが要求するプランの推定を行なう。このキーワードとは日常語であり、ユーザが日常使用している言葉を表す。また代表語とは、キーワードとの意味的関連性を記述するものであり、シソーラスで表現されている。またプラン知識ベース中のノードとは、プラン知識ベースの終端記号、非終端記号を表す。図8にキーワードと代表語とプラン知識ベースのノードとのマッピング情報であるキーワード知識ベースの例を示す。

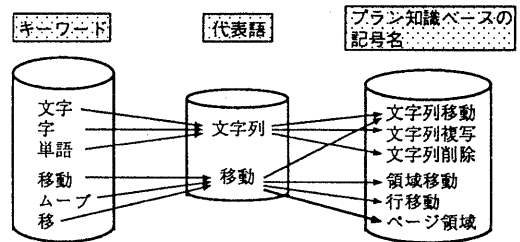


図8. キーワード知識ベースの例

複数の代表語が検索された場合は、代表語とノード間の参照回数（リンク数）が最大のノードを抽出する。

いま、ユーザの質問文が“文字を移動したい。”であった場合、キーワードとして“文字”と“移動”が選ばれる。それらの代表語は“文字列”、“移動”であるので、それらの指すリンク数が最大のものからユーザの要求するノードを得る。ここでは、“文字列移動”が選ばれる。

#### 3.2 プラン生成部

プラン生成部とは、キーワード検索部で得られたノードに至るまでのパスについて求める。いまかりにこのノードをユーザのゴールとし、以下このゴールに至るまでのパス生成について説明する。パス探索のアルゴリズムを図9に示す。

```

if (活性弧中にゴールが存在) (1)
    最新活性弧からゴールまでのパスを格納
else if (ゴールが活性弧の左辺ノードに到達可能) (2)
    最新活性弧からゴールまでのパスを格納
else if (活性弧の左辺ノードがゴールに到達可能) (3)
    最新活性弧からゴールまでのパスを格納
else (4)
    最新活性弧から開始記号のルールまでのパス1を格納
    開始記号からゴールまでのパス2を格納
    パス1にパス2を付加したものをパスとして格納
end if

```

(注) 開始記号: プラン知識ベースのルートとなる左辺ノード

図 9. パス探索のアルゴリズム

(1) 操作をすべての活性弧に対して行なう。次に(2),(3),(4)に対しても同様に行なう。パスが見つかるまで上記の処理を行なう。到達可能かどうかのチェックは、最左到達の到達可能行列を用いるのではなく、間接到達(最左到達以外のものに対する到達度)を付加した拡張到達可能行列を用いる。この拡張到達可能行列は、最左到達のものを~、間接到達可能のものを~'とした場合、図10を用いてパラメータ化したものであり、パラメータの値が小さいほど到達度は高くなる。これによりパス探索の処理時間を短くすることができる。

0	~	6	~', ~' ~
1	~'	7	~', ~ ~'
2	~' ~	8	~ ~' ~'
3	~ ~'	9	~' ~' ~'
4	~' ~'	10	others
5	~ ~' ~		

図 10. 拡張到達可能行列のパラメータ

状態解析パーサによるパス結果(活性弧のリンク情報)を図11に、拡張到達可能行列を図12に示す。ユーザのゴールを"文書作成"とする。まず図9中の(1)の操作を行なう。ゴールは活性弧(1)中に存在するので、最新活性弧(3)の<ファイル名の指定>から活性弧(1)の<文書作成>に対してのパスを調べる。到達可能行列図12より、到達度(パラメータの値)は2である。最新活性弧(3)を不活性弧にするノード列を求める。ここでは、まず(実行キー)がパスとして得られる。次に最新活性弧(3)の左辺ノードである<ファイル名の指定>を右辺に持つルールを抽出する。この際、到達度

2以下のノードを左辺にもつルールを抽出する。ここでは、活性弧(2)であるので、活性弧(2)について調べる。活性弧(2)の左辺ノードからゴールまでの到達度は1である。<ファイル名の指定>は、すでに求めたパス(実行キー)によってパスできるので、活性弧(2)を不活性弧にするノード列(<ファイル名の指定>の右側のノード列)を求める。<ファイル名の指定>の次ノードは存在しないから、同様にして、<ファイル保存>を右辺に持つルールを抽出する。ここでは活性弧(1)となるので、活性弧(1)について調べる。活性弧(1)の左辺ノードがゴールであるので、この活性弧を不活性弧にするノード列(<ファイル保存>の右側のノード列)をパスとして付加する。ここでは、<印刷>(終了)が得られ、パスとしては、(実行キー)<印刷>(終了)が得られる。

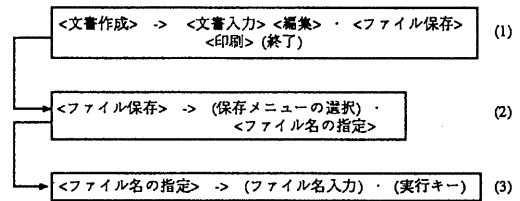


図 11. 活性弧のリンク情報

	文書作成	文書入力	編集	ファイル保存	ファイル名の指定
文書作成	0	10	10	10	10
文書入力	0	0	10	10	10
編集	1	10	0	10	10
ファイル保存	1	10	10	0	10
ファイル名の指定	2 (2,4)	1	10	1	0

(注) 複数のパラメータが存在する場合、最小の値をとる

図 12. 拡張到達可能行列

パスが複数存在する場合は、パスの長さ、ユーザモデルの値からユーザにとって適切なパスを決定する。

### 3.3 メッセージ生成部

プラン生成部から得られたパスは、終端記号、非終端記号の列として表される。メッセージ生成部は、この終端記号及び非終端記号から記号名を抽出し、文体パターン、メッセージ生成規則を用いてメッセージを生成する。この文体パターンとは、メッセージの述部のメッセージパターンが格納されているものである。またメッセージ生成規則とは、記号名と文体パターンとの関係を表すものである。図13に文体パターン、図14にメッセージ生成規則を示す。終端、非終端の記号名を直接メッセージに用いる為、メッセージを格納する必要がない。

文体パターン1 : "の実行方法を説明します。"  
 文体パターン2 : "を入力して下さい。"  
 文体パターン3 : "まず"  
 文体パターン4 : "を行なって下さい。"  
 文体パターン5 : "は終了しました。"  
 文体パターン6 : "は"  
 文体パターン7 : "で行ないます。"

図13. 文体パターン

```
if(パスのノード数 == 0)
    メッセージ文3
if(パスのノード数 == 1)
    メッセージ文1
else
    メッセージ文2 (1)
if(ノード1が未知 && ノード1が非終端)
    メッセージ文4 (2)
メッセージ文1:
    ゴール 文体パターン1 ノード1 文体パターン2
メッセージ文2:
    ゴール 文体パターン1 文体パターン3 ノード1 文体パターン4
メッセージ文3:
    ゴール 文体パターン5
メッセージ文4:
    ノード1 文体パターン6 ノード2* 文体パターン7
* : 0個以上
```

図14. メッセージ生成規則

パス中に、非終端でかつユーザにとって未知なノードが存在すると、モジュール側からユーザに対して詳細なメッセージが存在することを知らせる。ユーザがそれに対して説明を要求すると、非終端は展開され展開された内容についてメッセージが生成される。

いまプラン生成部で生成されたパスを(実行キー) <印刷> (終了) とし、ゴールを<文書作成>とす

ると、メッセージ生成部は、図14のメッセージ生成規則(1)のゴールに"文書作成"、ノード1に"実行キー"を代入することにより、"文書作成の実行方法を説明します。まず実行キーを入力して下さい。"というメッセージを生成する。

また<印刷>のユーザモデルが未知の場合、<印刷>を展開する。<印刷>は(印刷メニューの選択)で展開される。図14のメッセージ生成規則(2)のノード1に"印刷"、ノード2に"印刷メニューの選択"を代入し、メッセージ文"印刷は印刷メニューの選択で行います。"を生成する。

## 4 アクティブガイド

ユーザにとってアプリケーションのすべての機能を使いこなすことは大変困難である。そこで本システムは、常にユーザの操作をモニターし、ユーザが非効率なプランを実行した場合、積極的に効率的なプランを提示するものである。ユーザのモニターは、前述の状態解析パーサが行なう。

アクティブガイド用のルールの場合、パースされる際に、条件文の評価をし、それが真のときにメッセージを表示するものである。条件文が無い場合は、真として取り扱う。

ユーザがカーソルの位置を1ページの(1, 3)から(1, 2)、(1, 1)の位置に移動したとすると、状態解析パーサによって図14のパース結果を得る。アプリケーションからの処理情報により、変数は以下のように設定される。

ページ1 = 1、行1 = 1、列1 = 2、ページ2 = 1、行2 = 1、列2 = 1

次に、条件文を評価してみると真であるので、以下のメッセージを提示する。

- " shift キーと削除キーを同時に押すとカーソルを文の先頭へ移動できます。"

## 5 評価

事務系の社員10名について本システムの評価を行なった。対象のアプリケーションは日本語ワープロを用いた。評価方法は、被験者に簡単な文書をマニュアルを使って作成する場合と、ガイドモジュールを使って作成する場合の2つに分けてプロトコル

<カーソルを文書の先頭へ移動>

-> (shiftキーと削除キーを同時に押す)

-> (カーソル移動) (カーソル移動 ページ1 行1 列1)

(カーソル移動 ページ2 行2 列2) .

[ページ2==1 && 行2==1 && 列2==1

&& (行1!=1 || 列1!=1)]

{"shiftキーと削除キーを同時に押すと

カーソルを文書の先頭へ移動出来ます。"}]

図1 4. アクティブガイド用のルールのパース結果

解析を行なった。被験者10名の内、初心者3名と中級者は7名であった。以下その評価結果について示す。

- ガイドモジュールを使用することによって、マニュアルを使用するよりも2割程度の時間短縮が得られた。
- パッシブガイドの有用性の確認  
被験者すべてに対して必要な情報を提示してくれると答えた。
- アクティブガイドの有用性の確認  
被験者の8人に対して、必要な情報を提示してくれると答えた。
- 質問文入力方法に不便さを感じるユーザがいた。

質問文入力方法に不便を感じたユーザは、かな漢字変換に不慣れだった為と考えられる。

## 6 まとめ

今後、アプリケーションが複雑になるにつれて、ユーザにとって使いやすくまたユーザの習熟を促す機能はますます必要になってくると考えられる。そこで、我々は、現在のユーザインタフェースの抱える問題を明らかにし、この問題を解決する1つの方法としてパッシブガイドとアクティブガイドの開発を行なった。

本モジュールでは、ユーザに対してメッセージを提供するだけであるが、今後アニメーションを用いたガイドの提供も行っていきたい。

## 7 参考文献

[Aho 72] Aho, A.V. and Ullman, J.D., The Theory of Parsing, Translation, and Compiling, vol.1 and 2, Prentice-Hall, 1972.

[Desmarais 87] M.C. Desmarais, The Diagnosis of User Strategies, INTERACT'87, Elsevier Science Publisher B.V., 1987, pp186-189.

[Green 85] M. Green, Report on Dialogue Specification Tools, Springer-Verlag, Berlin, 1985, pp9-20.

[Kay 80] Kay, Algorithm Schemata and Data Structures in Syntactic Processing, TR CSL-80-12, Xerox PARC, Oct, 1980.

[経田 91] 経田 樹朗, 森 俊也, 北野 幹浩, 知的なガイド機能を応用ソフトウェアに提供するモジュールの開発, 情報処理学会第43回全国大会, 1991, vol.6, pp291-292.

[Kyoden 92] Kyoden, IGM: Intelligent Guidance Module, To appear in PRICAI'92, 1992.

[吉良 88] 吉良他, 知識ベースを用いたヘルプシステム, 情報処理学会第37回全国大会, 1988.

[宮崎 89] 宮崎他, プロダクションシステムに基づいたユーザインタフェース管理システム, 文書処理とヒューマンインタフェース, 1, 1989.

[Platt 73] Pratt, V.R., A Linguistic Oriented Programming Language, Proc. of IJCAI '73, pp372-381.

[Platt 75] Pratt, V.R., Lingol-A Progress Report, IJCAI '75, pp422-428.

[Tomita 86] M Tomita, An Efficient Parsing for Natural Language, Kluwer Academic Publishers, 1986.

[Tomita 87] M Tomita, An Efficient Augmented Context-Free Parsing Algorithm, Computational Linguistics, 13, 1-2, 31-46, 1987.

[永田 87] 永田他, 個別利用者モデルに基づく対話的ソフトウェアのメッセージ生成, 電子情報通信学会論文誌, D, J-70D 11, 1987.

[斉藤 87] 斉藤他, 情報構造に基づくユーザインタフェースマネージャの試作そのシステム構造, 情報処理学会第35回全国大会, 1987.

[鈴木 88] 鈴木, ユーザフレンドリ. アシスタンス-知的検索型マニュアル- 文書処理とヒューマンインタフェース, 20-4, 1988.