

連想型オンラインマニュアル Pop-Doc の 5L-9 構成とその評価

押見正雄 田中正樹 岡部啓

(株)CSK総合研究所

1. はじめに

近年、ソフトウェアの開発効率の向上のために、プログラミング環境の改善が図られている。開発の際に、マニュアルなどの文献を参照する機会は多く、この参照の手間を軽減する手段の一つに、オンラインマニュアルが挙げられる。しかし、従来のオンラインマニュアルは、マンマシン・インタフェースの点で問題が多い。そこで本論文では、その問題点を明らかにするとともに、我々の開発した連想型オンラインマニュアル Pop-Doc について述べる。

2. 従来のオンラインマニュアルの問題点

オンラインマニュアルの必要な機能は、通常の紙上のマニュアルの機能が代行でき、さらにより快適なマンマシン・インタフェースを実現していることである。マニュアルの重要な機能の一つである「検索」という観点から、従来のオンラインマニュアルの問題点を以下に述べる。

(1) 検索に手間がかかる

従来のオンラインマニュアルは、コマンドラインベースのものが多く、検索したいキーワードをキーボードから入力しなければならない。

(2) 検索可能な状況が少ない

通常のマニュアルは、手元にさえあれば、比較的どんな状況においても利用可能である。しかし、オンラインマニュアルは、エディタの中から使えないなどと、その利用に制限を受けることがある。

(3) キーワードの識別が簡単にできない

ある文字列を検索し、それがマニュアルに記載されていない場合、検索に要した手間は無駄となる。従って、検索以前にキーワードの識別が容易に行なえることが望ましい。

3. Pop-Doc の概要

あるキーワードの情報をマニュアルから得ると、その中に再び検索したいキーワードが存在することがある。このようとき、従来のオンラインマニュアルでは、ディスプレイ上にキーワードが表示されているにもかかわらず、再度、キーボードより入力しなければならなかった。従って、人間とコンピュータとの情報のやり取りの中に無駄が生じていた。連想型オンラインマニュアル Pop-Doc は、このようにある情報から別の情報を引き出すような連鎖的行為を、無駄なく行なうことを可能にする。

本システムの構成図を図1に示す。現在、我々は32ビットのパーソナルコンピュータ上で動作する Amorphous Prolog/Lisp 処理系を開発中であり、Pop-Doc はこの処理系上で利用することを前提として開発された。また、マンマシ

ン・インタフェースを向上させるため、デバイスドライバ型のマルチウィンドウシステムである Domino Window と、オンラインマニュアルと一体型のエディタ AmE を開発した。開発にはアセンブラとC言語を用いた。

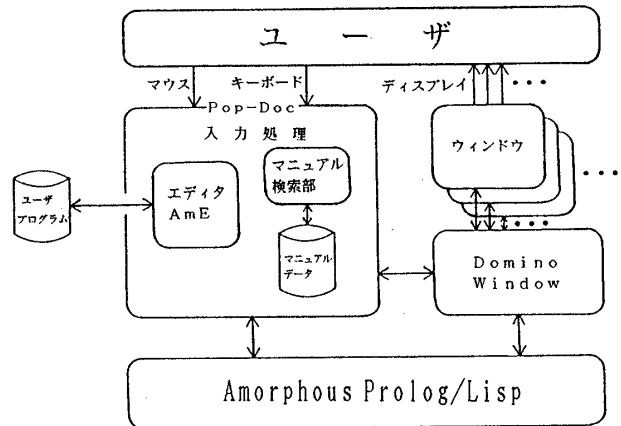


図1 システム構成図

4. 特徴

(1) キーワードスキャナー型

従来のシステムでは、ディスプレイやウィンドウの特定の領域にあらかじめ機能を設定しておき、マウスなどのポインティングデバイスによって、各機能の選択を行なってきた。しかし、本システムでは、マウスが自分自身の前後の文字を調べ、特定のキーワードが存在するならば、そのキーワードの示す文字列オブジェクト(後述)に、メッセージを発行するという手法で実現した。このとき、パラメータとしてウィンドウ情報やマウス情報を渡している。

この手法を用いると、エディタなどのようにキーボードから入力されたキーワードや、エラーメッセージのように処理系が出力したキーワードに対しても、容易にマニュアルを参照することができる。

具体的な動作は、図2のように検索可能なキーワードの上にマウスが侵入すると、そのキーワードが反転する。そして、マウスをクリックすることによって、マウスの指す位置に存在する全てのキーワードが表示される。図3は、図2より表示されたマニュアルのテキスト上で、'writeq' という文字列をさらに検索しようとしているところである。

(2) キーワードの入力の手間の軽減

マウスによって、ディスプレイ上の文字列を自動的に判別し、検索のためのキーワードとして指定できるので、キーボードから改めて入力する必要がなくなった。

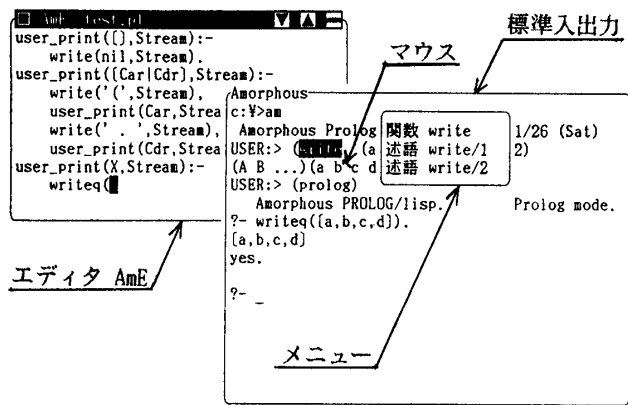


図2 動作例 (1)

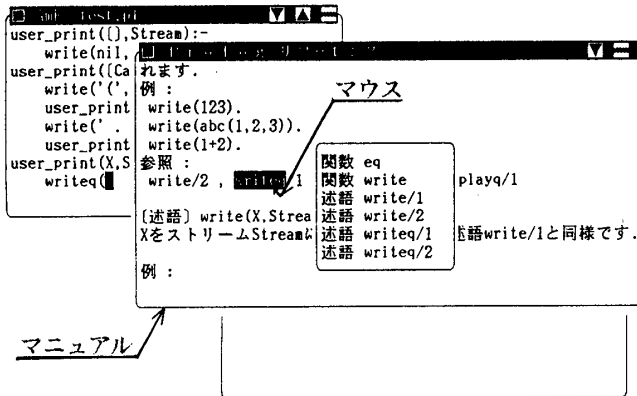


図3 動作例 (2)

(3) キーボードの入力時にも検索可能

Pop-Doc は、Amorphous Prolog/Lisp 処理系上のコンソール入力時に使用できるので、検索したい単語をすぐに入力し、マウスを用いて参照する事ができる。

(4) キーワードの存在の識別

図2のようにキーワードの存在するところだけが反転するので、ある文字列がキーワードかどうかをすぐに識別できる。

(5) エディタ AmE との一体化

マニュアルはプログラム編集に参照されることが多い。そこで、常時 Pop-Doc を利用できるエディタ AmE を開発した。このエディタは、マニュアルとそのインデックスを動的に構築する機能を持つ。例えば、編集時の Lisp プログラムの '(defun)' の位置より、各関数の定義位置を自動的にピックアップし、インデックスを作成する。このインデックスを用いてユーザ関数名より、そのプログラムを自動的に表示することができる。この機能によって、比較的容易に他人のプログラムを解釈できると考えられる。

(6) Domino Window

Pop-Doc は、Domino Window 上で動作し、複数のマニュアルをマルチウィンドウによって同時に参照できる。このウィンドウシステムの開発によって、ウィンドウ内で折り返されたキーワードも検出できるようになり、マンマシン・インタフェースが向上した。

(7) その他

従来のオンラインマニュアルにある文字列サーチ機能やページめくり機能も実現した。

5. 文字列オブジェクト

Pop-Doc では、文字列オブジェクトという概念を導入した。文字列オブジェクトは、それ自身がメソッドとデータを持ち、マウスからのメッセージによって、それらのメソッドが起動される。

マウスは、自分の指す位置から前後に範囲を広げながら、キーワードを探索する。そして、特定のキーワードが発見されると、そのキーワードに相当する文字列オブジェクトを生成し、これにメッセージを送る。通常、このオブジェクトは、マニュアルにインデックスされているので、マウスがクリックされたときに表示されるメニューオブジェクトに自分自身を登録する。また、その他のオブジェクトは、予め登録されたメソッドを起動する。例えば、'CLOSE' という文字列に、ウィンドウを消去するメソッドを登録しておけば、特定のリージョンを設定することなく、ディスプレイ上の 'CLOSE' という文字列をマウスで触れただけで、ウィンドウの消去を行なうことができる。

このように、文字列オブジェクトは、マウスによって引き起こされる動作に多様性を与えることができる。

6. 評価

まだ当研究所内の評価ではあるが、エディタ AmE 上で Pop-Doc を使用することによって、ソフトウェア開発の効率が向上することが確認された。特に、Prolog, Lisp, C などのリファレンス・マニュアルをオンライン化しておくことが、開発効率の向上につながるようである。

7. 課題

(1) 図や表の取り扱い

現在、本システムの表示データは、テキスト表現のみに限定しているが、図や表を取り扱う必要がある。今後は、図や表を表示するオブジェクトを導入し、対処していく予定である。

(2) マルチフォントへの対応

現在、Pop-Doc は単一フォントにしか対応していないが、マルチフォントへの拡張を予定している。問題点としては、マウスの示す地点の文字列の切り出しが複雑になったり、ウィンドウシステムが大規模なものになってしまうことが考えられる。

(3) マニュアル入力ツールの充実

マニュアル原稿の入力やインデックスの作成を容易に行なうためには、以下のようなマニュアル入力ツールを充実させていく必要がある。

- ・ インデックス作成ツール
- ・ テキスト整形ツール
- ・ キーワード自動抽出ツール
- ・ マニュアル構成管理ツール

8. おわりに

本研究開発を行なうに当たって、貴重な助言を与えてくださったCRIの所員の方々に感謝致します。