

6L-8

技法・意図を備えた プログラム作成支援システム

藤波 努 芳賀 博英

(株)日立製作所 システム開発開発研究所

1. はじめに

本システムの目的は、プログラム作成に利用できる情報を整理・蓄積し、それを参照することで初心者プログラマーが熟練プログラマー並みにプログラムを作成できるようにすることである。先に我々はプログラム作成において利用される情報を「プログラミング技法」「プログラムの意図」「ソースコードテキスト」の3つに分類し、相互に関連する情報間にはリンクを張るプログラム情報管理方式を提案した¹⁾。本報告では、このプログラム情報管理方式を採用したHI-UX PROLOG (以下Prologと記す)用プログラム作成支援システムの概要と、ワークステーション2050上に作成したプロトタイプシステムの実現方式について述べる。

2. プログラム作成支援システムの機能

今回開発したプログラム作成支援システムは、ハイパーテキスト²⁾の概念を応用した情報管理機能と、プログラム実行機能を持っている。

2.1 情報管理機能

本システムは、以下の情報を管理している。(図1)

(1) フラグメント

本システムでは個々のプログラム情報をフラグメントと呼ぶ。フラグメントにはプログラム情報の分類に従って、「プログラミング技法」「プログラムの意図」「ソースコードテキスト」の三つの種別がある。

(2) リンク

リンクは、特定のフラグメントまたはフラグメントの任意の部分と関連するフラグメントを指示する情報であり、リンク先のフラグメントの種別に従って三つの種別があるほか、次に述べる「上位」「下位」リンクがある。また利用者が任意のリンクを設定することもできる。

(3) ワールド

同じ種別のフラグメントの集合は「一般-特殊」関係により階層化し、すぐ上のフラグメントに対しては「上位」リンクを、すぐ下のフラグメントに対しては「下位」リンクを設定して管理する。

これらの情報に対して、フラグメントの編集、リンクの設定・削除ができる。またリンクをたどることにより、関連するフラグメントを検索することができ、①ある「ソースコードテキスト」に使われている「プログラミング技法」と「プログラムの意図」を検索する、②ある「プログラミング技法」を使った「ソースコードテキスト」の例を検索する、③ある「プログラムの意図」を実現している「ソースコードテキスト」の例を検索することなどができる。

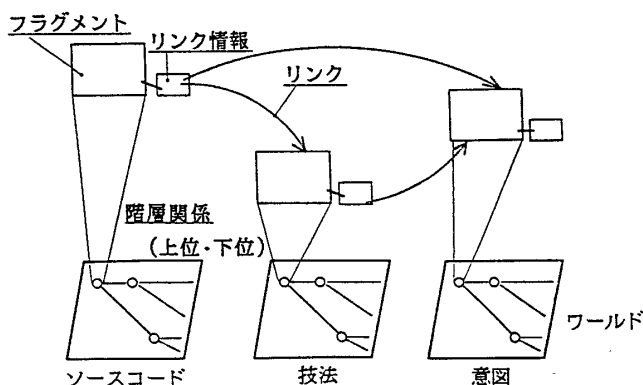


図1 プログラム情報管理方式のイメージ

2.2 プログラム実行機能

検索した「ソースコードテキスト」フラグメントを直接実行し、迅速にプログラムの動作を確認することができる。

3. ユーザインタフェース

(1) フラグメント (図2)

「ソースコードテキスト」の場合はソースコードを、「プログラミング技法」「プログラムの意図」の場合は内容を記述した文書を、そのフラグメントから張られているリンク種別の名称とともに表示する。リンク種別名称を選択すると、そのリンク種別のリンク情報を表示する。

(2) リンク (図2)

選択した種別のリンクで関連を指示しているフラグメントの名称とコメントを表示する。フラグメント名称を選択するとそのフラグメントを表示する。

Programming Support System with technique and intention,

Tsutomu Fujinami, Hirohide Haga,

Systems Development Laboratory, Hitachi, Ltd.

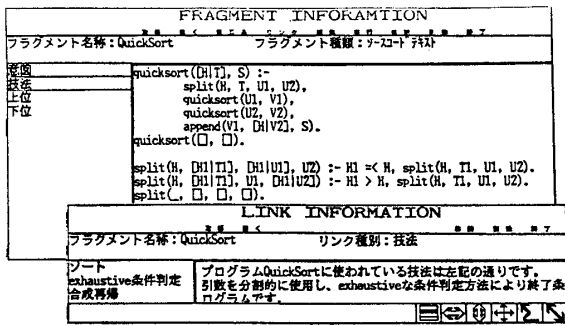


図2 フラグメントとリンクの表示

(3) ワールド (図3)

「プログラミング技法」「プログラムの意図」「ソースコードテキスト」の各ワールドごとに階層構造を解析し、フラグメント名称をノードとするネットワーク構造を表示する。フラグメント名称を選択することにより、フラグメントの表示/非表示ができる。

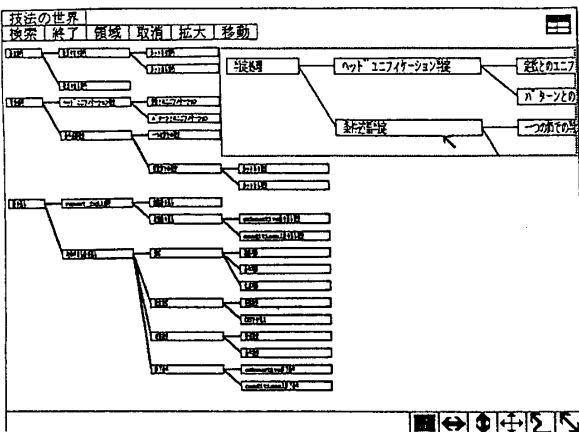


図3 ワールドの表示

4. システム構成

本システムは図4に示すようにマルチプロセスで構成している。プロセスは「フラグメント制御プロセス」「ワールド制御プロセス」「Prolog実行プロセス」とそれらをコントロールする「システム制御プロセス」の4種類があり、個々のフラグメントとワールドはそれぞれ1つのプロセスを割当てている。次に各プロセスの機能について述べる。

(1) フラグメント制御プロセス

他プロセスでの処理に従って、フラグメントの表示/非表示をおこなうほか、編集作業、リンク情報の表示などをおこなう。

(2) ワールド制御プロセス

各ワールドのフラグメントとリンクを解析して階層構造を表示するほか、フラグメントの修正を動的に反

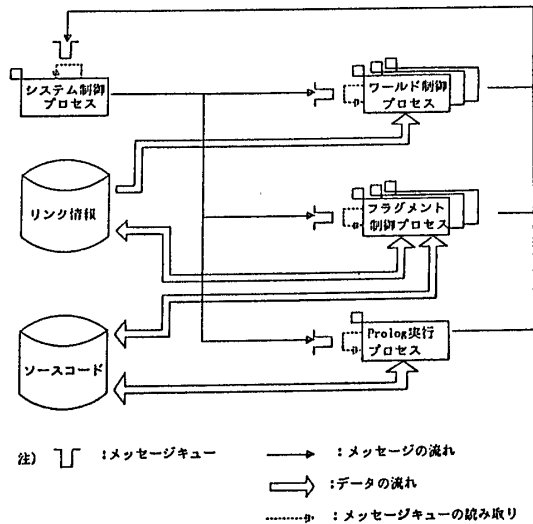
映して表示を変化させたり、マウス選択したフラグメントの表示/非表示を受け付ける。表示画面は、領域を指定して拡大/移動することができる。

(3) Prolog実行プロセス

Prologインタプリタを制御して、指定されたソースコードをロードし実行する。

(4) システム制御プロセス

他プロセスの生成/消滅を管理し、各プロセス間のメッセージ交換を制御する。他に自分自身のウィンドウからユーザ入力コマンドを受け付けて、指定されたフラグメントやワールドの表示、ソースプログラムの編集、指定されたプログラムのPrologインタプリタへのロード、HI-UXのOSコマンドの実行をおこなう。



注) □ :メッセージキュー → :メッセージの流れ
 ⇨ :データの流れ
 :メッセージキューの読み取り

図4 プロセス構成

5. まとめ

提案したプログラム情報管理方式を採用したProlog用プログラム作成支援システムのプロトタイプシステムをワークステーション2050上に開発した。本システムは、利用者がソースコード以外にプログラミング技法やプログラムの意図を利用できるので、参考となるソースコードが見つからない場合でも、プログラミング技法を参考にプログラムを作成できるなど応用性が高いものである。

参考文献

1) 芳賀他: プログラム作成支援システムにおけるプログラム情報管理方式, 日本ソフトウェア科学会第5回大会論文集, pp33-36(1988)
 2) J.Conklin: "Hypertext: An Introduction and Survey", IEEE COMPUTER, pp17-41(Sep.1987)