

IS-5

プログラミング・ガイド・システム

— システム概要 —

藤野かおる 中川由佳里 飯塚史之

株式会社 富士通研究所

1. はじめに

プログラミングには、OSに関する知識から、プログラム言語の文法規則に関する知識、アルゴリズムに関する知識、対象分野に関する一般的な知識まで幅広い知識が必要である。さらに、熟練プログラマは経験的にエラー時の対処方法、効率の良いプログラムの書き方、陥りやすい間違いの回避方法などのプログラミング・ノウハウを豊富に持っている、このプログラミング・ノウハウが質の高いプログラムを効率良く作成するために重要な役割を持つと考えられる。このような観点から、我々はプログラミング・ノウハウを含めたプログラミング知識を利用して、プログラム作成を支援するプログラミング・ガイド・システムの開発を行っている。本稿では、本システムの特徴、構成、インタフェースについて報告する。

2. システムの特徴

本システムでは、文法に関する知識を持ち、簡単なプログラムが作れるレベルの中級プログラマを対象として、エディタ上での試行錯誤的なプログラム開発を支援する。エディタ上でプログラムを編集するユーザは、マニュアルを調べたり、以前の似たようなプログラムを探して参考にしたり、一部を修正して使ったり、熟練プログラマに聞いたりして問題を解決していくと考えられる。従来の試行錯誤的なプログラム開発においては、このような「調べる」「人に聞く」「思い出す」「覚える」「記述する」「確認する」などの煩わしい作業で思考を中断されるという問題点があった。この点を解決するために、本システムでは文法、プログラム、ノウハウを体系化した知識ベースを作成し、これを対象とした検索/登録機能、及び知識ベースを利用した記述支援、構造表示、ソース・プログラムの検証機能などを提供する。

3. システム構成

プログラミング知識を体系化した知識ベース、検索サブシステム、登録サブシステム、検証サブシステムが本システムの主要構成要素である。

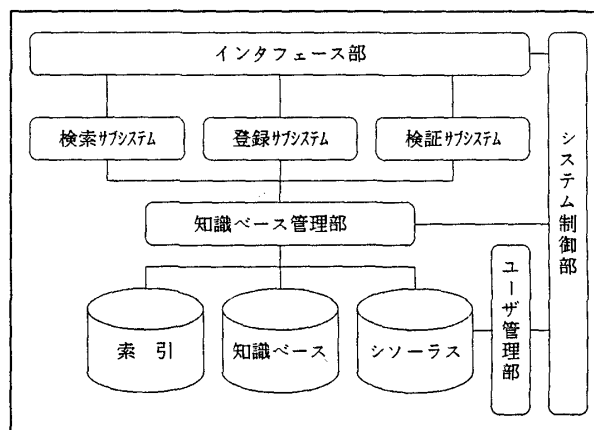


図1. システム構成図

3.1 プログラミング知識

本システムでは、プログラミング知識として、マニュアルに記載されているプログラム言語の文法規則、用語の説明、一部を修正すれば利用できるプログラム部品、参考になるプログラム、熟練したプログラマが経験から獲得するノウハウを扱う。ノウハウとして次のようなものを含む。

- (1)ある要求をプログラム上で実現するための手段や手順
- (2)ある要求をプログラム上で実現するための条件や規則
- (3)ある概念の属性や機能の説明

3.1.1 知識の枠組み

プログラミング知識には、概念を説明するものから手順や方法などを示すメタ知識までが含まれる。また、知識の表現形式にも、文章、ソース・プログラム、ルールなどのばらつきが見られる。プログラミング知識を扱うためにはこのようにレベル・大きさ・内容の詳細さ・種類・表現形式にばらつきがある知識を上手く整理し、かつユーザが知識を追加/参照し易い知識構造にする必要がある。本システムでは、フレームとリンクから構成されるネットワーク構造を用いて知識を整理している。

まず、個々のフレームを次のように作成する。知識を概念や機能ごとの扱いやすい単位に分割する。この知識の種類に応じて用語、プログラム、ノウハウの3種類の

カテゴリに分類する。一つ一つの知識に対して主題を表す複数のキーワードを付ける。次に、知識の内容をフレーム内のスロットを用いて整理する。スロットには共通スロットとカテゴリ別スロットがあり、共通スロットにはキーワード、識別番号、知識名などを格納する。カテゴリ別スロットにはソース・プログラム、自然言語による文章、条件部と実行部からなるルールなどで表現される知識の本体を格納する。作成したフレームは知識ベースに格納する。フレーム化した知識の間には、類似、上位下位、部分と全体などの関係がある。また、キーワードとして使用される語の間にも同義などの関係がある。このような知識間の関係やキーワード間の関係をリンク情報としてシソーラスに格納する。キーワードと、そのキーワードが付けられた知識の間のリンク情報を索引に格納する。

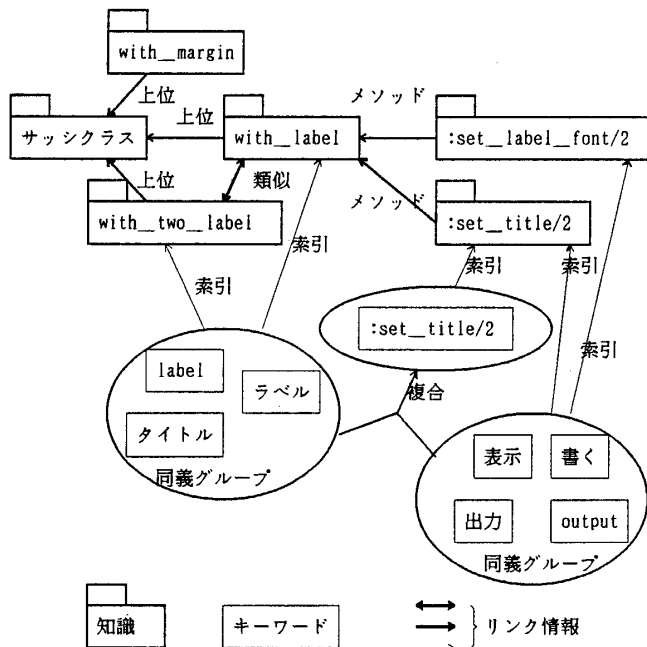


図2. 知識構成図

3.1.2 知識の管理

効果的なプログラミング支援のためには、知識ベース内に有益な知識を多く持つ必要がある。そこで、システムでは基本の知識ベースを用意すると共に、ユーザ自身が知識ベースを拡張していくための登録機能を提供する。知識にはシステム提供/他ユーザ登録/自己登録の登録ユーザ区分があり、全ての知識が参照可能である。また、自己登録知識のみの修正・削除が可能である。その他、マルチ・ユーザの同時アクセス制御、テンポラリファイルを利用した知識ベースの保護、修正時間短縮のための全ユーザログアウト時の知識ベース一括修正などの機能を持つ。

3.2 登録サブシステム

編集中のプログラムの一部やノウハウの知識ベースへの登録、シソーラス修正を支援する。

3.3 検索サブシステム

キーワードと論理演算子からなる検索論理式により知識ベースから知識を取り出す。また、シソーラス内のリンク情報を用いた関連知識の検索を行う。

3.4 検証サブシステム

知識ベース内の知識を利用して、プログラム構造の表示、誤りのある箇所指摘、記述支援などの様々な形でプログラム作成を支援する。

(1) 構造表示機能

参照、被参照、継承、被継承等の関係に応じたプログラム構造を図示する。

(2) 関数ガイド機能

陥りやすいプログラム上の間違い（スペルミス、構文ミス、効率の悪い箇所等）を指摘し、改良方法を指示したり、未使用の関数を紹介する。

(3) 注釈文管理機能

スプレッド・シートを利用した注釈文の記述支援、変数、行、ブロックを単位とした注釈文の階層管理、表示・非表示の切り換え、注釈文の抽出等を行う。

4. ユーザ・インタフェース

システム環境として以下のような基本的なインタフェースを実現する。

(1) X-windowを用いたマルチウィンドウ環境

(2) コマンドとポップ・アップ・メニューの併用

(3) エディタ、登録/検索サブシステム間のカット・アンド・パッファ機能

5. おわりに

本システムは、SUN ワークステーション上にLISPでインプリメントしている。知識ベースには、ESP (Extended Selfcontained Prolog)のウィンドウに関連した知識を収集した。今後は、システム上での検索/登録/検証機能の実験・評価を行うと共に、インタフェースの充実、ノウハウを利用した支援機能の拡張を行っていく予定である。

[参考文献]

中川, 他: プログラミング・ガイド・システム—検索サブシステム—, 情報処理学会第39回全国大会論文集, 19