

プログラミング質問応答システム Lisp-PALの実現と評価

2K-1

西岡里枝子, 上原 三八, 小川 知也
(株) 富士通研究所

1. はじめに

近年、ソフトウェアの需要が増大するに従い、ソフトウェア開発の生産性の向上と開発要員の教育が重要な課題となっている。我々はこの課題に対応し、プログラマの知識を活用した質問応答システム Lisp-PAL (Lisp Programming Assistant for Learners) を開発した。

Lisp-PALは、Lispプログラミングの初心者を対象に、日本語での問い合わせに対して、日本語とメニューを併用してプログラム例と説明を中心としたプログラミング知識を提示する。我々は、オブジェクト指向言語による知識表現とオブジェクトのネットワークの操作技術、及び、単語間の意味/関係付け情報を用いた自然言語解析技術を考案し、これら技術により実現した諸機能について既に報告した〔1, 2〕。

本稿では、Lisp-PALのシステム概要、実用上の問題点とそれらを解決した技術、及び、モニタによる評価結果について報告する。

2. システム概要

システム構成を図1に示す。本システムは以下の機能を持つ。

- ①日本語による質問応答機能
- ②日本語による記述からフレーム構造を生成する知識ベース生成機能
- ③日本語による知識追加機能
- ④未知語/句登録機能
- ⑤任意の目次に沿って知識全体を提示する機能

以上の機能により、プログラミング知識をプログラミング中に手軽に問い合わせできるようになった。

図2にFMR上でのシステムの使用例を示す。図の下側のウィンドウでプログラムを編集し、上側のウィンドウでシステムとの質問応答を行っている。システムが提示した知識を利用し、プログラムを編集している様子を示している。

図3に知識の記述例を示す。上記②の機能により、知識を内部実現方法によらず記述できるため、従来のフレーム記述言語で記述する場合と比較して知識の登録と保守が飛躍的に容易になった。

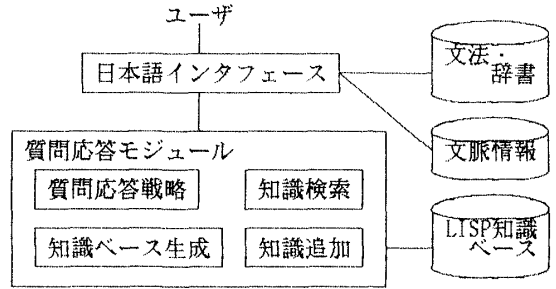


図1 システム構成図

```

リストをつなぐには?
リストをつなぐ関数には NCONC と APPEND があります。
リストをつなぐノウハウを知りません。
次のノウハウは関連していませんか? 番号で指定して下さい。
1 dolist によるリスト処理
2 リストをスタックに使う方法

appendの構文は?
関数APPENDの構文は、「append &rest {リスト}*」です。

GNU Emacs 21.4.7 3:55pm 12 May 89 Lisp 9/13: null (q)home > Ctrl-Shift
; (oblis #'((a b c) (d) (e f))) --> (a b c d e f)
;

(defun oblis (vec)
  (let ((result '())
        (i (length vec)))
    (loop
      (when (= i 0) (return result))
      (setq result (append

```

図2 システム使用例

```

; *****
関数「append」の
機能は「リストをつなぐ」。
説明は「引数のリストからなるリストを新たに作ります。引数のリストは
変化しません」。
構文は「append &rest {リスト}*」。
使用例は「> (append '(a b c) '(e d f) '() '(g))
(a b c e d f g)
> (append '(a b c) 'd)
(a b c . d)」。
GNU Emacs 21.4.7 5:21pm 12 May 89 Lisp 0:YMSBYVLMYALFUC:RUB

```

図3 知識記述例

3. 実用上の問題点とその解決技術

(1) 階層化されたキーの利用

本システムは、「配列を作るには」等のプログラミングしたい内容での問い合わせ（機能検索）を実現している。この様な検索では、日本語解析の結果そのもので検索しても、目的とする知識が検索できない場合が多い。これは、問い合わせ文に名詞と動詞の自由な組み合わせ

が起こるために、その表現の自由度が高まることによる。我々は、知識ベース中の検索対象となる日本語文をその意味により階層的に表現しておき、問い合わせ文の意味より下位の階層関係にあるものを意味的に含まれると解釈して、知識を検索する方式を実現した。図4に本方式の原理を示す。本方式によると、関連知識や代替の機能を実現する知識についても応答することができる。

(2)知識ベースのロードの高速化

知識を日本語で記述したことで記述性は向上したが、逆に、知識ベース生成/ロードに時間がかかり、システム起動が実用に耐えられなくなった。この問題は、図5に示すように、知識ベースのデータ構造をLISPプログラムの形に変換し、さらにこれをLISPコンパイラでコンパイルしておく方式により解決した。これにより、知識のロード時間を1/30程度に短縮できた。

4.システム評価

社内モニタによるシステムの評価を行った。評価は、モニタからのアンケートと全質問応答のログ情報をもとに行った。

(1)システム全体

システムの有益性は認められたが(図6)、日本語でどのように質問するかに戸惑う点、HELPメッセージやガイドの不備等のインタフェースの問題点が指摘された。

(2)質問応答能力

モニタの質問の約半数に正しく答えられず、その原因は知識不足と単語不足が85%を占めた(図7)。回答されない場合、ユーザは日本語解析ができないと思い込み、言い回しを少し変えて繰り返し質問する傾向が認められるので、回答できない理由をメッセージに明確に含めることが必要であることがわかった。

(3)日本語インタフェースの有効性

LISPプログラミングの初級者の56%が、「質問応答システムには日本語インタフェースだけで充分」と回答したが、中上級者の44%は不充分と回答し、メニューやコマンドの併用を望んだ。

(4)知識ベース全体の提示

目次に沿って知識ベース全体を提示する機能は、初心者にはシステム主導で知識を参照する手段として、中上級者には知識ベースの内容を手軽に知る手段として利用され、有効であることがわかった。

5.おわりに

評価の結果、ユーザの経験によりユーザインタフェースへの要求が変化することを確認した。

今後は、システムがプログラミング作業の支援ツールとして有効かどうか等の観点から評価・改良を行う予定である。

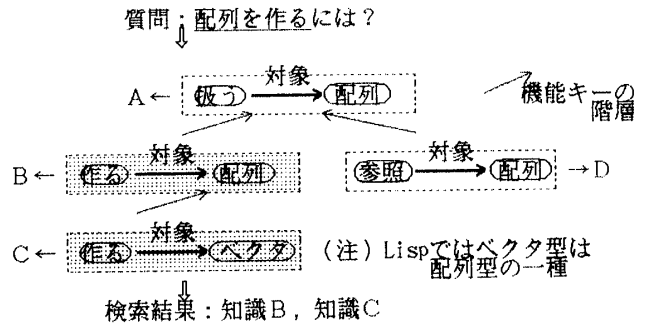


図4 機能検索方式

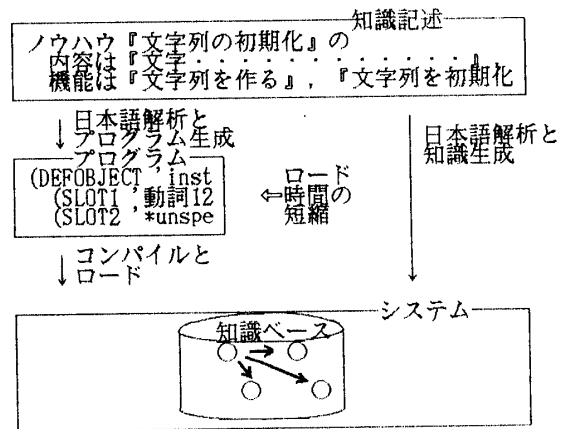


図5 知識ベースのロードの高速化

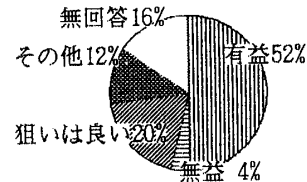


図6 システムの有効性

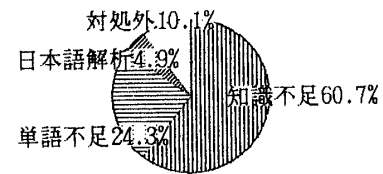


図7 回答できない原因

〔参考文献〕

〔1〕S.Uehara, R.Nishioka, T.Ogawa, and T.Mohri, "Implementing Programming Consultation System LISP-PAL, Framework for Handling Frame-Based Knowledge in Object-Oriented Paradigm," CIIAM(1986), pp.409-423
 〔2〕上原, 西岡, 小川, 毛利, "LISPプログラミング知識に基づく質問応答システム," 日本ソフトウェア科学会第4回大会(1987), pp.367-370