

7N-2

プログラム作成支援システムにおける 知識に関する検討

吉田 敦 田中 英彦 元岡 達

(東京大学 工学部)

1. はじめに

知識工学とは、人工知能に関する研究の成果を、実社会における種々の問題解決に用いる為の研究分野と考えられている。しかしながら、現在の所、知識工学という言葉だけが先行し、その定義は曖昧なままである。また、知識工学の応用である知識ベース・システムについても、その定義ははっきりしておらず、知識ベース・システムの構築法も、確立されていないのが現状である。

本研究では、知識ベース・システムの構築法を提案することを目的として、二次元図形入出力プログラムを対象とする知識工学に基いたプログラム作成支援システムを試作してきた。本稿では、このシステムの持つ知識の分析と、それに基く考察・検討について述べる。

2. モデルの設定

本システムでは、プログラミングを、次に述べるような操作として考えている。この操作は、次の3つのステップに分けられる。

- ① 概略仕様の形成
- ② 概略仕様の詳細化
- ③ プログラムコードの生成

①においては、対象とするプログラムの、各機能を一つの英文で表わす。プログラムに対する概略仕様全体は、これらの英文の集まりで記述される。また、プログラムで用いるデータに関する記述は、このステップで決定されることとする。ここでは、対象とするプログラムで用いるデータとして、単一の定数、単一の変数、階層化されたりレージョンの集まりで表わされる構造データを考えている。

②においては、①で得た概略仕様と、データに関する記述を基に、概略仕様を段階的に詳細化していき、詳細な仕様を得る。このとき、利用可能なサブルーチン・パッケージやライブラリの中のサブルーチンの利用、対象とするプログラムで用いるデータに関する記述の参照等を行なう。

③では、実装言語の文法等を考慮し、②で作られた詳細な仕様を、特定言語のプログラムコードに変換する操作を行なう。

3. プログラム作成支援システムで用いる知識

本システムは、2で述べたモデルを基本として構築されている。本システムでは、知識を、プログラミング一般に関する知識、応用領域固有の知識、実装言語固有の知識に分けて保有している。ここでは、これらの知識を、2のモデルに沿って、概略仕様形成の為の知識、仕様の詳細化の知識、プログラムコード生成の為の知識に分けて考えてみる。

3.1 概略仕様形成の為の知識

概略仕様の形成では、プログラムの入力した仕様を見て、必要だが欠けている部分を補う為の知識が必要である。本システムでは、この為の知識として、図1に示すような知識を実装している。

図1. プログラミング一般に関する知識

```
refinement([subst,Var1,Var2]):-
    is_defined(Var2),
    !,
    make_dspect1([subst,Var1,Var2]).

refinement([subst,Var1,Var2]):-
    !,
    insertdef(Var2),
    make_dspect1([subst,Var1,Var2]).

refinement([read,Data]):-!,
    refinement([prompt]),
    refinement([readinputdevice,
                Data,Inputdev]),
    refinement([errorcheck,Data]).
```

ここで実装されている知識は、プログラムの入力中にある機能を実現する為に必要な前処理が指定されてなければ、これを適切な位置に追加することを言明している。このような知識が、プログラムの大まかな機能毎に実装されており、対象領域を拡大する為には、このような知識を新たに追加する必要がある。

3.2 仕様の詳細化の為の知識

本システムでは、仕様の詳細化の基本操作として、以下のような操作を考え、これらの操作に対応する知識を実装してある。

1. 与えられた処理の記述を、複数の処理のシーケンスに置き変える。
2. 与えられた処理、または、詳細化で得られた処理シーケンス中の処理を実現するサブルーチンの選択と、その呼び出し手続きの実装

3. 手続き選択の為に必要なパラメータの推論

4. その他

図2にその例を示す。これらの知識の用いる基本操作は、概略仕様の参照、プログラムで用いるデータに関する記述の参照、サブルーチンに関する記述の参照、プログラマへの問い合わせ、詳細仕様の形成等である。

プログラミング一般に関する知識、応用領域固有の知識いづれについてもこのような分類ができる。

図2. 応用領域固有の知識

```
refinement([specify,[Varx,Vary]]):-!,
  varrole(Varx,[specify,position]),
  varrole(Vary,[specify,position]),
  typeofdisplay(raster),!,
  refinement([readlocator,Varx,Vary,
    dragging]).

refinement([draw,polygon,Xarray,Yarray]):-!,
  variables(Xarray,_,[S],_,_),
  variables(Yarray,_,[S],_,_),
  !,
  makearraylast(Xarray,S,Xarrays),
  makearraylast(Yarray,S,Yarrays),
  refinement([call,movea,
    [Xarrays,Yarrays]]),
  refinement([call,pllina,
    [Xarray,Yarray,S]]).
```

3.3 プログラムコード生成の為に知識

3.2の知識による操作で最終的に得られた詳細仕様は、実装言語固有の知識により、特定の実装言語のプログラムコードに変換される。この為の知識は、例えば、図3のようになる。図3を見てわかるように、これは、詳細仕様中の特定パターンと、これに対応する、特定言語のステートメントの系列の対応を表わしている。

図3. 実装言語固有の知識

```
fortran_conv([while do,Cond,Statelist):-
  fortcondrev(Cond,Fcond),
  generate_label(Lab1),
  writeln([' ',Lab1,' ',continue]),
  generate_label(Lab2),
  append(Fcond,[goto,Lab2],Temp1),
  append([' ',if],Temp1,Temp2),
  writeln(Temp2),
  fortranconv0(Statelist),
  writeln([' ',go,to,Lab1]),
  writeln([' ',Lab2,' ',continue]).
```

4. 考察と検討

ここでは、2で述べたモデルの問題点を述べ、このモデルに基いて実装された、3で述べた知識について、考察・検討をしてみる。

4.1 モデルに関して

ここで考えた、プログラミングに対するモデルは、極めて制約の強いものである。たとえば、概略仕様として与える記述では、必ず、一つの機能を一つの英文で記述する、プログラムで用いるデータとして、

2に挙げたものしか許さない、詳細化をtop-downに行なう等の制約がある。このようなモデルが、他の応用領域向けのプログラムでも通用するという保証は全くない。従って、対象とするプログラムの範囲の拡張をするとき、プログラミングに対するモデルの変更が必要になってくる可能性が大きい。

従って、対象とする問題（プログラミングの支援）のモデリングや、モデルの修正を支援するツールは、必要なシステム要素であるといってよい。但し、このようなツールについては、まだ構想の段階である。

4.2 本システムの知識に関して

プログラム作成支援の対象領域を拡大・変更しようとする、その為に、知識の追加が必要になる。このような、知識の追加に関しては、プログラミングに対するモデルが変わらない限りシステムの制御部分は変更しなくてもよい。しかし、次のような問題点がある。

本システムでは、知識は、いくつかの基本操作から成る。効率よい拡張のためには、システムに新たに追加される知識は、これらの基本操作のみを用いて記述されるべきである。システムのユーザがこの条件を満たしつつシステムを拡張するためには、知識の追加を誘導する、知識追加支援ツールが必要であるが、本システムにはこのようなツールがない。従って、効率よく知識を追加することが、困難になっている。

5. 終わりに

本研究では、プログラミング支援を対象領域として、知識ベース・システムの構築法を考察してきた。このシステムは、限られた範囲のプログラムを対象とするシステムではあるが、十分な拡張性が期待されるものである。また、このシステムの試作を通じて、プログラム作成支援に必要な知識に関する知見も得られた。しかし、システムの拡張を容易にするには、前述のようなツールが必要であり、これらのツールの実現には、解決されるべき問題も多い。今後、これらのツールに関する基礎的な研究の必要がある。

<<参考文献>>

[1] 吉田、田中、元岡：「知識工学の手法を用いたソフトウェア作成支援システム」第27回情報処理学会全国大会4B-9 pp529 ~ 530

[2] 吉田、田中、元岡：「プロダクションルールを用いたプログラム作成支援システムに関する考察と検討」第28回情報処理学会全国大会3G-9 pp 1055~1056