ソフトウェア工学 67－5
（１９８９．７．２５）

部品の使用を前提としたプログラムの保守支援エキスパートシステム

花岡 晃浩　門倉 敏夫　深澤 良彰

早稲田大学　理工学部

本システムは、部品を用いて作成されたプログラムの機能変更保守の支援を目的とする。部品の機能はあらかじめ分かっているため、その知識を本システムに用意しておくことで、保守に必要な情報の一部を省略できる。さらに、ユーザが保守要求を本システムに入力する際、その部品の知識を利用することで、ユーザの負担を軽減する。つまり、ユーザは対象プログラムについての情報を完全に把握していなくてもよい。そして、保守要求に該当する変更部分が部品である場合のみ、本システムは、ユーザに部品中の変更箇所を指示する。これによって、ユーザの望む保守を、比較的簡単に、高品質でおこなうことが可能となる。

An Expert System Assuming the Use of Building Blocks for Supporting Maintenance

Akihiro HANAOKA, Toshio KADOKURA and Yoshiaki FUKAZAWA

School of Science and Engineering, Waseda University, 3-4-1, Okubo Sinjuku-ku, Tokyo 169, Japan

In this paper, a system which intends to support the maintenance of functional modification for target programs used building blocks is described. This system can reduce a part of the information for maintenance by using knowledges of the function of building blocks. Also, the utilization of its knowledge makes the user description of maintenance requirements easier. Therefore, the user does not recognize the detail structure on the target program. This system shows the changing point in the building blocks to the user, only when the changing points which satisfy the maintenace requirements is in the building blocks. Accordingly, we can deal with the desired maintenance more easily in the high quality.

1. はじめに

ソフトウェア工学では、仕様記述やプロトタイピングなど様々な研究がなされ、その進歩も著しい。しかし、保守に関する研究についてみたとき、保守はライフサイクル中で最も重要といえるにもかかわらず、その研究はあまり進歩していない[1,2]。その理由としては、保守が後向きで地味な作業であること、保守の方法論に決定的なものがないことなどがあげられる。

一般的な「保守支援ツール」は、プログラム関連図や木構造チャートなどといった、保守に必要な情報をユーザに提示する[3,4]。しかし、そのようなツールは、変更すべき箇所の特定までは行なわないのが現状である。

以上の点から、保守に役立つ方法論の研究と、保守の際に変更すべき箇所の特定を行うツールの開発の必要性は高いといえる。

1．1 目的

本システムの目的は、ユーザからの保守要求を満たすように、プログラム中の変更箇所を指示することである。しかし、ソースコードを解析し、保守を支援する方法は、プログラムの意味の解析等の点で困難である。本システムでは、部品を使用したプログラムの保守を前提とする。そのために、部品の知識をあらかじめ用意し、それを保守に利用することとする。したがって、修正すべき箇所が部品中にない場合は、本システムは保守の支援を行えない。また、場合によっては、変更箇所の特定を行うために、ユーザと対話を行う必要が生じる。このために、グラフィックなども利用できるＥＷＳ(Engineering Work Station)環境を利用する。

これによって、ユーザの望む保守を、比較的簡単に、高品質でおこなうことが可能となる。

1．2 前提

本システムの設計・試作においては、以下の諸条件を前提としている。

(1) 保守内容

保守の内容は、修正、適応、完全化、更新保守に大別できる。これらの中で、本システムは、更新保守における、機能変更保守を対象とする。

(2) 保守対象

対象プログラムは部品を用いて作成されているものとする。ただし、その全てが部品で構成されていなくても、また、部品の一部が変更されていても良い。

(3) 使用部品

早稲田大学における学内事務処理システムの構築のために定義されたWISDOM (Waseda Information Systems: Development and OperationMethod) [5]における部品ライブラリを仮定する。WISDOMにおける部品の種類としては、部品中のサブルーチン内の詳細な処理やデータの記述等を、部品使用者が行うホワイト・ボックス部品と、ユーザが部品内部に干渉できないブラック・ボックス部品の２つがある。本システムでは、この内のホワイト・ボックス部品を対象とする。そこでは、事務処理ソフト用のパッチ・パターンとして、抽出、編集等１０種類ほどの部品が用意されている。

この部品は構造化プログラミングの可能なCOBOL/Sで記述されている。

(4) 保守対象プログラムの構造

対象プログラムの全体が、機能単位でモジュール化され、そのモジュール毎に順次的に処理されるようなものであるとする。つまり、対象プログラムが、機能単位のレベルでは、モジュール間で再帰の構造をとっていない（図１参照）。

(5) 実現環境

本システムは、NEC-EWS4800 のエキスパートシステム構築シェルＥＸ－ＣＯＲＥ上で実現している。

2. 本システムの概要

本システムの情報の入出力の概略を以下に示す。

(1) 部品に関する知識の入力

本システム内に用意される部品の知識は、フレーム表現とルール型表現で記述する。

(2) 対象プログラムの入力

本システムに保守対象プログラムの情報を入力する。稼働状態にあるプログラムの情報として、つぎの二つの入力が必要である。

① 対象プログラムのプログラム網図

(Program Network Diagram：以下ＰＮＤ図と略す) ＰＮＤ図の例を図１に示す。

② 部品が改造されて使われている場合の改造造情報

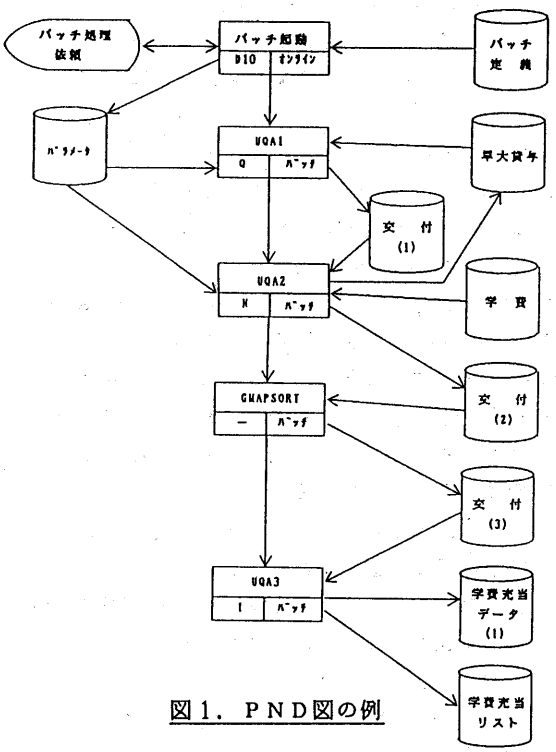

図１．ＰＮＤ図の例

(3) 保守要求入力

ユーザから保守を要求された場合に、保守要求は簡易言語または対話型で本システムに入力される。

(4) 保守要求該当部分の指示

保守要求に対応する変更箇所が部品の場合に、本システムは部品中のプロシージャ等の変更すべき箇所をユーザに指示する。

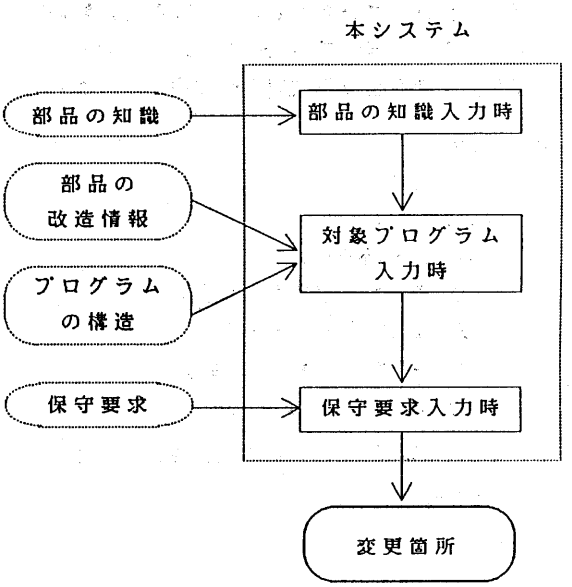

図２．本システムの入出力の概略

３． 部品の知識記述

部品の知識は、ルール型表現とフレーム型表現を用いて行う。保守を支援するために必要となる部品の知識は、つぎの３種から構成される。

(1) メタ知識

部品はいくつかの機能を持ち、各機能はいくつかのプローシージャから構成されている。ここにおけるメタ知識とは、一つの機能がどのプローシージャから構成されているかを記述するものである。

(2) 状態変化

ある一つの処理を行うことによって、部品中の状態が変化することを記述する。

(3) プローシージャ間の関係

プローシージャ間の処理の前後関係に依存する部分や、プローシージャを実行する際に必要な状態を記述する。

例として、入出力ファイルを各１つずつ持ち、データ中のキーに応じて、大・中・小計処理を行い、整形して出力する部品（編集部品）の記述の一部を表現形式に分けて、以下に示す。ただし、この部品の記述は全体で約５０行である。

```
(instance
    (name I01)
    (class 部品)
    (instance-variable
    (処理 報告書（明細書）作成)
    (INPUT-FILE 入力ファイル１)
    (OUTPUT-FILE 出力ファイル１)
    (小計処理 (SMALL-SUM-PROC))
    (中計処理 (I11S,MIDDLE-SUM-PROC))  }(a)
    (大計処理 (I13S,LARGE-SUM-PROC))
            :
    ))
```

図３．部品のフレーム表現（編集部品の例）

```
IF (:EXECUTE FILE-OPEN)
  THEN (:STATE ファイルオープン)
IF (:EXECUTE ﾃﾞｰﾀ読み込み処理)   }(b)
  THEN (:STATE 新データ入力)
          :
```

図４．部品の前向きルール表現（編集部品の例）

```
IF (:STATE リスト・ヘッダ印字)
  THEN (:EXECUTE 集計結果印字)
IF (:BXECUTE ファイルオープン)    }(c)
  THEN (:EXECUTE データ読み込み処理)
          :
```

図５．部品の後向きルール表現（編集部品の例）

（ａ）の部分は中計処理を行うためには、中計処理のサブルーチン(MIDDLE-SUM-PROC)、および、集計キーの変化を調べるルーチン(I11S)を実行する必要があることをフレーム表現で表している。

（ｂ）の部分はデータ読み込み処理のサブルーチンを実行すると新データ入力状態になることをルールを用いて表現している。あるサブルーチンの結果生じる変化を記述するために前向きのルールを用いている。このルールは、前向きに利用することをコンパイル時に指定する。

（ｃ）の部分はデータ読み込み処理のサブルーチンを実行するためには、ファイルオープンのサブルーチンを実行しなければならないことを意味するルールである。これは、あるサブルーチンを実行する際に、先行して実行する必要があるサブルーチンを記述するために、後向きのルールを使用している。このルールは、後向きに利用することをコンパイル時に指定する。

４．保守対象プログラムの構造の認識

構造の認識において必要なことは、データの流れとモジュール間の接続関係である。事務処理のプログラムにおいては、ファイルやリストを一つのデータ構造と考えることができる。そこで、ＰＮＤ図から、入力データ（ファイル等）や新たに生成されたデータを認識することができる。

さらに、部品の知識を用意しているため、部品から出力されるデータについての情報も正確に得ることが可能である。例えば、編集の処理を行う部品から出力されるデータは入力データの大・中・小計処理の結果であるといったような情報がこれにあたる。これによって、データの接続関係を明確にすることができる。

また、ＰＮＤ図を用いることによりモジュールの接続関係の認識も容易である。

以上のような、データの流れと処理部分の知識とＰＮＤ図とから、与えられたプログラムの構造を分析する。

本システムでは、ＰＮＤ図の情報をフレーム型表現で持っている。

図１のＰＮＤ図中の一部分の表現を以下に示す。

```
(instance (name UQA3)
          (class 処理)
          (instance-variable
            (処理種類 ﾊﾞｯﾁ) (部品ﾊﾟﾀｰﾝ I01)))
```

図６．部品のインスタンス表現

これはUQA3というモジュールが編集部品（部品パターンＩ０１）から構成されるバッチ処理部品であることを示している。

このインスタンス表現は以下のようなモジュールのフレーム表現から生成される。

```
(class (name 処理)
       (instance-variable
              (ｻﾌﾞｼｽﾃﾑ名 nil) (ｼﾞｮﾌﾞ名 nil)))
```

図７．ＰＮＤ図中のモジュールのフレーム表現

５．部品の改造情報

実際のプログラム中では、部品は通常改造されて　使用される。この場合、あらかじめ用意された部品の知識との食い違いが生じる。そこで、部品の改造についての情報が必要となる。

ファイルやリストの数の変更は、ＰＮＤ図から直接に認識できる。その他の改造情報は部品改造記述言語により記述される。

例えば、図３に示した編集部品を図１で使用する際には小計処理が不要である。このこと、すなわちモジュール名ＵＱＡ３という部品から小計処理を削除することを本システムの部品改造記述言語では以下のように表現する。

（削除　UQA3　小計処理）

この表現は、ＥＸ－ＣＯＲＥの持つ述語型表現を用いている。

６．保守要求の入力法

保守要求が、あるファイルにおけるレコードの抽出の条件を変えたいといった比較的簡単なものであるならば、本システムが用意している保守要求記述言語で表現する。保守要求に必要な情報をユーザが完全に把握していない場合や、本言語では表現が困難な場合には、ユーザと本システムとが対話的に保守要求を決定していく。

６．１ 保守要求記述言語で入力する場合

例　ファイルＡの内容をリストＡとして印字して欲しいという保守要求の場合は、次のように記述する。

（after　ファイルＡ　append-new-list
　リストＡ　as　output-of　ファイルＡ）

６．２ 対話的に入力する場合

ユーザの持つ保守要求の鍵を得るために、本システムは、ユーザに対し、キーワードの入力を求める。次に、本システムは、そのキーワードを含むＰＮＤ図をユーザに提示する。この時、ユーザが、ＰＮＤ図中のモジュール等を指示すると、本システムは後述の浅い知識を用いて、指示された部分に関するオペレーションをユーザに提示する。それを参考にして、ユーザは保守対象とそのオペレーションを指示し、保守要求を具体的なものとしていく。

浅い知識とは、部品全体が行う処理に関する知識である。例えば、「抽出」の部品の場合には、一つの入力ファイルからある条件に合うデータを抽出し、新しいファイルをつくるという機能を持つという知識がこれにあたる。この知識は、対話的に保守要求を決定する際に用いられる。つまり、この知識を使って、ユーザの抱く保守要求をより具体的なものにする。

７．本システムにおける推論過程

本システムの推論過程を以下に述べる。

(1) 保守に必要な情報の構築

本システムは、ＰＮＤ図をフレーム表現に変換し、蓄積する。また改造情報によって、あらかじめ用意されている部品の基本的知識を、現実の部品の内容に対応した知識に変換する。

(2) 本システムの動作

保守要求が入力されると、本システムは図６のような過程で推論をおこなう。本システムの推論法は、人間の保守の熟練者を参考とし、推論過程を、以下の三つのフェーズに分割している。

① 保守要求分析フェーズ

ユーザから与えられる保守要求を分析して、

システムの内部表現に変換するフェーズ。保守要求中の情報が不足の場合は、ユーザにＰＮＤ図等を提示して、ユーザが不足分を補う。

② 対象プログラム分析フェーズ

（1）の保守要求の内部表現を元に、該当部分のチェックを行い、その部分の変更が他に影響を与えるかどうかを分析し、より具体的な保守要求に変換するフェーズである。つまり、このフェーズにより、モジュール単位の保守要求を決定する。

③ 保守部分指示フェーズ

該当するモジュールが部品の場合のみ、本システムは、ユーザに変更部分を指示する。また、その変更が他に影響を与える場合は、ユーザに確認をとり、保守要求に沿う改造箇所を指示する。その後、本システムは行なわれた保守に基づいてプログラムの情報を書き換える。

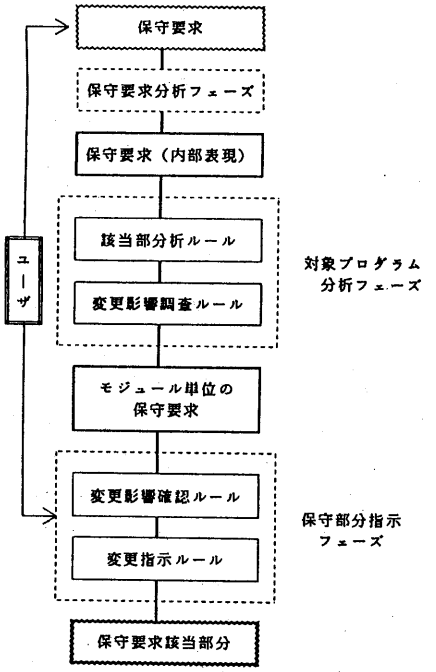

図8．本システムの推論過程

8．評価

保守要求記述言語の記述能力を評価するために、早稲田大学の事務システムで実際に生じた１１個の保守要求に対して記述を試みた。その結果、９個は記述することができた。しかし、その内の２個については、自然言語による保守要求記述のみでは情報不足であった。残りの２個については、現実の処理内容に関わり過ぎており、記述が不可能であった。

9．おわりに

本システムは、現在、NEC-EWS4800上で、エキスパートシステム構築シェルＥＸ－ＣＯＲＥを用いて作成中である。できるだけ早くシステムを完成させ、実際の保守作業に適用し、その効果を確かめていく予定である。

今後の拡張として、保守が要求されているプログラムのソースコードを入力すれば、自動的にそのプログラムの構造を分析するようなシステムを用意することを考えている。

謝辞

この研究のために多くのデータを提供してくださった、早稲田大学事務システムセンターの方々に心より感謝致します。

参考文献

[1] G.Parikh :
"Handbook of Software Maintenance,"
John Wiley & Sons, Inc, 1986.

[2] N.F.Schneudewind :
"The State of Software Maintenance,"
IEEE TRANS. ON S.E., VOL. SE-13, No.3,
March 1987.

[3] D.J.Reifer and S.Trattner :
"A Glossary of Software Tools and Techniques,"
Computer, July, 1977, pp.52-60

[4] N.Zvegintzov : "Nanotrends,"
Datamation, August, 1983, pp.106-116.

[5] 白井　克彦，東　基衛 :
「事務システム標準化マニュアル」，
日刊工業新聞社，1985.