

TM-0245

プログラム自動生成システム

岩元莞二, 西谷泰昭, 和田 孝
(日本電気)

December, 1987

©1987, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

プログラム自動生成システム

Automatic Program Generation System

岩元 亮二*
Ryōji Iwamoto

西谷 泰昭*
Yasuaki Nishitani

和田 孝*
Takashi Wada

要 旨

プログラマ向けに書かれたモジュール仕様書は、日本語の文章、図、表の組合せで記述されているものが多くみられます。本稿では、このような人間にとって書き易く理解し易い自然な形式で書かれたモジュール仕様書を理解して自動的にプログラムを生成する実験システム PGEN を紹介します。

現在、実験用の小規模な辞書を用いていますので、使用できる語いは制限されていますが、意味的な曖昧さや省略のある表現もある程度理解してプログラムを生成することができます。

Usually, module specifications given to programmers are written in the combination of natural language (Japanese), figures, and tables. This paper introduces experimental system PGEN which understands such specifications and automatically generates computer programs.

Although a restricted number of vocabulary is allowed in the current system because of the experimental small scale dictionary, the system can understand, to some extent, semantically ambiguous and abbreviated expressions and produces programs as the result.

1. まえがき

プログラム自動生成は古くて新しいテーマであり、絶えず1950年代の FORTRAN や ALGOL に始まるプログラム言語の高級化の流れと、1960年代に始まったアルゴリズムの自動合成の研究の流れがあります。高級言語化の流れは問題向きの超高級言語、仕様記述言語、あるいは第4世代言語の形で発展し続けていますが、高級化はどうしても汎用性とのトレードオフの問題を含んでいます。一方、自動生成という意味合いで最も期待されているアルゴリズムの自動合成は、仕様からプログラムへの変換 (Program

transformation) 技法を中心に研究されていますが、まだ基礎研究の段階で実用になるまではなお多くの研究を要すると思われます。

最近ソフトウェア開発のさまざまな局面に対し効率化のためのさまざまな工夫がなされており、その中でプログラム自動生成が種々の形で盛り込まれています。たとえば、ソフトウェア生産システムに組込みの自動化機能の例として、設計記述からのプログラムスケルトン生成、既存プログラム部品を活用したプログラム合成などがあります。この場合はプログラム全体が自動生成されるわけではありません。部分的に自動生成されたものに人間が手を加えて完成したプログラムに仕立て上げていく必要があります。完全自動化できない部分は日本語の文章で処理内容が書かれている所です。そこは人間が意味を理解してプログラム言語に変換しなければなりません。

日本語によるプログラミングもこれまでいろいろ試みられてきましたが、機械的に処理し易いようにプログラム言語と同じような強い制限がついていたため書きにくく実用には到りませんでした。

本稿で紹介するプログラム自動生成システム PGEN は、実際に使用されているモジュール仕様書を参考にして制限の少ない自然な形式で記述された仕様を理解してプログラムを自動生成するシステムです (写真)。

現状のプログラム仕様書は、その設計意図をプログラマに伝達するとともに、保守、ユーザによるレビューなどプログラマ以外の人々にとっても共通の理解のベースとなっております。そのため、主に自然言語 (日本語) で記述されており、簡潔さや分かり易さのため図表も多用されています。

PGEN の目標は、このような人間向けに書かれた仕様書を理解することであり、それによって仕様作成者に新たな負担をかけずにプログラミング作業を自動化することです。

このような機能を実現するために自然言語処理、知識情報処理などの技術を利用しています。

* ソフトウェア生産技術研究所 生産技術開発部
Software Product Engineering Laboratory



写真 プログラム自動生成システム PGEN
Photo Automatic program generation system PGEN.

PGEN の開発では前に述べましたように仕様書の理解が主な技術課題であり自動翻訳と類似していますが、適用対象をモジュール仕様書に限定して意味解析を十分に行う点で違います。

たとえば、複数の解釈が考えられる曖昧な表現や部分的に省略のある表現でも、業務用語やプログラミングに関する知識と人間の経験的解釈規則を利用して正確な解釈を行うように努めます。

2. システム構成と機能概要

PGEN は図 1 のようなシステム構成をしています。システムの狙いは、自由度の高い日本語の文章と図、表で書かれたモジュール仕様を理解してプログラムを自動生成することです。まだ実験システムですが、基本的には以下の特徴を持っています。

- ① 日本語の文法上の制限を意識させない。
- ② 業務用語を用いた仕様記述ができる。
- ③ 使用できる用言（動詞、助動詞）は制限される。
- ④ ある程度の曖昧さや省略のある文も理解できる。
- ⑤ 日本語の文章と図、表を組み合わせることで意味を理解する。
- ⑥ 数式的な表現を埋め込んだ日本語文章を理解する。

以上の特徴により、使用できる語に制限はありますが、モジュール仕様書を書くには十分な自由度があります。

ただし、本システムを利用してプログラムを作成する前に準備作業が必要です。モジュール仕様書の中で使われている業務用語はプログラムではデータ名やデータの値になりますが、業務用語とデータの対応関係、およびデータ構造の設計結果は、辞書作成ツールを使ってあらかじめ変数辞書とデータ型辞書の中に登録しておく必要があります。

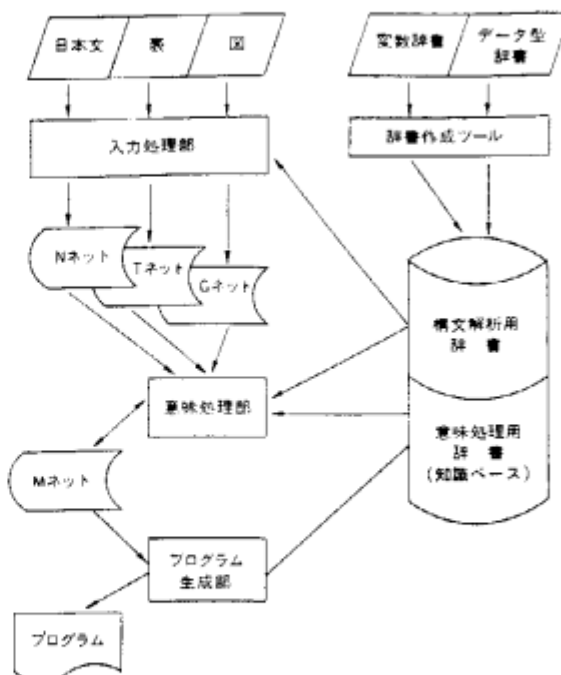


図 1 PGEN のシステム構成
Fig. 1 System structure of PGEN.

システムは対象ソフトウェアに固有なこれら 2 つの辞書をもとに構文解析用辞書と意味処理用辞書を修正し、それらを参照してモジュール仕様書を理解しプログラムを生成します。そのときの処理の流れは図 1 に示す通りです。

入力処理部は、仕様書作成用のエディタ機能と入力した構文解析機能を兼ねています。日本文、表、図の構文解析結果はそれぞれ、N ネット、T ネット、G ネットというネットワークで表現されます。意味処理部はこの 3 種類のネットワークを結合して仕様書の意味を表す M ネットを作成します。その後、M ネット上で冗長な情報の削除、意味的曖昧さの解消、省略語の補充などの意味処理を行って、プログラム生成に適した形になるように M ネットを変形します。プログラム生成部は M ネットをその構造に従って解釈しながら COBOL のプログラムを生成します。

3. モジュール仕様の記述例

本システムが処理できるモジュール仕様の記述例を図 2 と図 3 に示します。図 2 はプログラムの手続き部の処理内容を文章で記述したものであり、その中で参照されている表と図は図 3 に示したものです。

なお、データ宣言部は変数辞書から自動生成されます。変数辞書に変数定義を登録するときの記述形式は文章形式ではありません。COBOL のデータ宣言形式に近いものです。

以下では記述例に従って本システムが処理できる仕様記

- (1.) 入金総額の計算
金額が「00」でない入金明細について、以下を繰り返す
(1) 総額に入金額を加える
(2) 入出金口数をカウントアップする。
(2.) 監査キーコードエリアのセット
有帳のとき、以下の処理を行う。
(2.1.)
使用区分が1のとき、以下を行う
(1) コード数をカウントアップする
(2) コード数が5以下なら、コード数番目のコードを「321」とする
0のとき、後述以外なら、エラーコードを「321」として、リターン
(3.) 注意コードエリアのセット
有帳のとき、以下の処理を行う。
(3.1.)
使用区分が1のとき、以下を行う
(1) コード数をカウントアップする
(2) コード数が5以下なら、コード数番目のコードを「321」とする
0のとき、後述以外なら、エラーコードを「321」としてリターン。
(4.)
科目コードが「20」のとき、摘要コードに「00」をセットし、
「22」のときは、「N1」をセット。
(5.)
通帳がマル機程度額より大きいとき、表2に記し摘要コードに値をセット
してリターンする。
(6.)
流動性預金のとき、表1の検査値以外なら、エラーを409としてリターンする
(7.)
(2)のとき、リターンする
(8.)
図1に記し、値をセット

図2 日本語による仕様記述例

Fig. 2 An example of specification description in Japanese sentences.

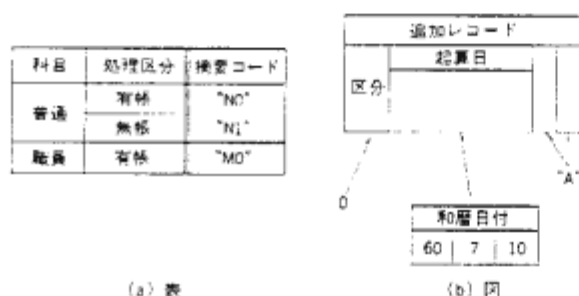


図3 表と図の例

Fig. 3 Examples of table and figures.

述の特徴を説明します。

(1) 節・段落構造

仕様記述は節・段落構造を持っています。節は節番号「(1)」と「(2)」で囲まれていると節タイトル(省略可能)からなる節ヘッダで始まります。節の本体は日本語文章や節の系列から構成されます。また、節の本体には(1)や(2)などで始まる箇条書きもできます。日本語による記述では、「以下を行う」や「以下の処理を行う」などの表現が良く出てきますが、「以下」がどの範囲かを理解するときに節・段落構造を利用します。

(2) 節タイトルの利用

節タイトルは通常はコメント扱いですが、節本体の記述での情報不足を補う役割も果たします。現在、本システムではデータの修飾子を見つけるのに利用しています。たとえば、図2の〔2〕節と〔3〕節で「コード数」という名詞が使われていますが、データ型辞書の内容と節タイトル

から、〔2〕節では監査キーコードエリアの、〔3〕節では注意コードエリアのサブフィールドの名前であると判断します。

(3) 処理条件の簡略表記

〔2〕節の「有帳のとき」は、正確には「処理区分が有帳のとき」ですが、データ型辞書を参照して「有帳」がコード表「処理区分」のコード値であることを知り、正しく解釈することができます。

(4) 主語や目的語の省略

「有帳のとき」でも主語が省略されていますが、〔4〕節の「22」のときは「では主語である科目コードが省略され、「N1」をセット」では目的語である摘要コード(に)が省略されています。この例の場合は文のパターンから省略語を補充できます。このほかにも、局所的な文脈から省略語を補充する場合があります。たとえば、「AとBを加え、Zに代入する」を「AとBの和をZに代入する」と変換してZに代入されるものをはっきりさせることができます。この例では、「代入する」という操作に関するプログラミング知識「代入されるものは数式でなければならない」に合致するような解釈を探っています。

(5) 表の参照

〔5〕節では条件の一部を文章で述べ、残りの組合せ的な条件は表を参照しています。また、処理内容の大部分は文章で述べていますが、どんな値をセットするかは表の中で示しています。本システムはこのような文章と表の意味を結合して完全な仕様として理解します。

(6) 図の参照

〔7〕節と〔8〕節は図を参照した例です。同一図でもそれを参照している文章が違えば図の意味も違ってきます。〔7〕節では条件を示すものとして理解し、〔8〕節では代入の関係を示すものとして理解します。また、図は完全なものでもなくとも、データ型辞書を見て不足情報を補うことができます。

(7) 数式混入の日本語文

図2の記述例にはありませんが、プログラムの仕様を表わす日本語の中には数式が日本語化した形で混入することがよくあります。その場合には、たとえば、「入力の通帳残高が元帳の通常残高でなければ・・・」のように数式の範囲が曖昧になったり、「N+1番目の入出金明細」のように構造的曖昧さ(「(N-1)番目の入出金明細」か「N+(1)番目の入出金明細」か)が生じ得ます。本システムはこのような曖昧さを日本文構文解析の段階で解決します。

以上のほかに、記述例にないような曖昧な表現や冗長表現もある程度処理できますが、詳細は省略します。

4. 仕様理解のための知識

PGENシステムは仕様理解とプログラム生成のためにさまざまな知識を使っています。それらは、プログラムに組み込みの知識とプログラムと分離した知識に分けられま

す。プログラムに組み込みの知識は主に日本語の文法に関する知識と、意味解析法に関する経験的知識です。プログラムから分離した知識は構文解析用辞書と意味処理用辞書に集約されますが、これらは事前に変換ツールを使って、日本語マスタ辞書、データ型マスタ辞書、変数マスタ辞書、Mモデル(辞書)から生成します。

以上の知識の中でプログラム生成に特有のものは、データ型辞書、変数マスタ辞書、およびMモデルです。特にMモデルは、対象ソフトウェアに依存しないもので、プログラムの構成要素(演算子、データ、式、関数、ループ構造、分岐構造、など)に関する知識を持っています。

図4はMモデルの一部を示したものです。Mモデルは、人工知能における代表的な知識表現法の1つである「フレーム」の考え方に沿って記述されています。フレームは型にはまった状況、概念、物などを表現する基本形を提供するものであり、表現の対象物は1つのフレームという枠の中で表現します。図4では各ノードがフレームに対応します。フレームの特性を定義するために属性を表わすスロットが必要な数だけ1つのフレームに割り当てられます。各スロットは属性名と属性値を持ちます。図4では、スロットはアークで示されており、アークのラベルが属性名、アークの終点のノードが属性値に対応します。属性名はフレームごとに定義できますが共通の属性名として「ako」があります。「ako」はa kind ofの略でスーパークラスとクラスの関係を示しています。図の例では、「add」(加算)が「exp」(式)の一種であることを示しています。すなわち「add」のスーパークラスは「exp」であることを示しています。

図4のMモデルは前述の例、「AとBを加え、Zに代入する」の意味解析に利用しています。Mモデルから、「代入する」に対応する概念「set」は、左辺(「obj_l」)と右辺(「obj_r」)を必要とすることが分かりますが、例文では左辺がZである点にははっきりしていますが、右辺は明確ではありません。図4のMモデルから、右辺は式でなければならず、また加算「add」は式の種類であることが分かります。この知識を、直前の文「AとBを加え」が加算を意味し得るという事実から、Zに代入されるものは「AとBの和」とであると解釈します。

このようにMモデルは曖昧な意味表現に対してプログラムとして妥当な意味解釈を選択するための手段を提供しています。

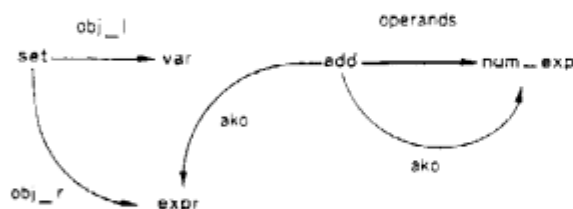


図4 Mモデルの一部
Fig. 4 Part of M-Model.

5. プログラム生成

モジュール仕様の意味処理結果は、Mモデルに具体的な値を付与したものから構成されるネットワーク(Mネット)で表現されます。プログラム生成部ではこのMネットを解釈してプログラムを生成します。このとき必要な知識はプログラミング言語についての知識ですが、PGENではMネットのノードに対応する用語にプログラム生成用の小さな関数を割り当てることによってその知識を持っています。たとえば、「eq」なる用語に対しては、「eqの左手の先のノードについてプログラムを生成し、「=」を出力せよ。次に右手の先のノードについてプログラムを生成せよ。」というような関数が割り当てられています。

一方、データ宣言部については、ユーザが定義した変数を参照しながら、COBOLの形式で出力します。

6. むすび

PGENは実験システムであり辞書類はまだ実験用の小規模のものです。いくつかの例題について入力からプログラム生成まで行えることを確認しました。

今後はさらに能力の安定化とレベルアップのための研究が必要と考えています。なお、本研究は、(財)新世代コンピュータ技術開発機構からの委託の一環として行ったものです。

参考文献

- 1) 岩元：「PGEN-1：仕様理解をベースとしたプログラム自動生成システム」、情報処理学会第32回全国大会、5M-6、pp.1271~1272、1986年3月。
- 2) 和田：「数式を含む日本文の構文解析」、情報処理学会第32回全国大会、4S-4、pp.1623~1624、1986年3月。

筆者紹介



Kanji Iwamoto

いわもと かんじ

岩元 寛二 昭和41年、東北大学大学院修士課程修了。同年、日本電気㈱入社。現在、ソフトウェア生産技術研究所生産技術開発部技術課長。電子通信学会、情報処理学会各会員。



Yasuaki Nishitani

にしに やすあき

西谷 泰昭 昭和56年、東北大学大学院情報工学専攻修士課程修了。同年、日本電気㈱入社。以来、ソフトウェア生産技術研究所にて、ソフトウェア工学、自動プログラミングの研究開発に従事。電子通信学会、情報処理学会各会員。工学博士。



Takashi Wada

わだ たかし

和田 孝 昭和58年、東京大学大学院理学系研究科情報科学専門修士課程修了。同年、日本電気㈱入社。以来、ソフトウェア生産技術研究所にてプログラム自動生成システム、とくに日本文処理の研究開発に従事。情報処理学会各会員。