

2F-7

日本語による仕様書からのプログラム自動生成

行徳孝彦 石川由美子 上原憲二

三菱電機 情報電子研究所

1. はじめに

最近、自然言語で記述された仕様文からプログラムを合成する研究が盛んになり、自然言語表現から仕様を獲得する各種の手法が提案されている。

我々は、制限の加わった日本語で書かれた設計仕様書より、手続き型プログラムを生成するプログラム自動生成システムの開発を行っている。本稿では、仕様書からの仕様獲得方法を中心に報告する。システムは、仕様書がもつ章や節の構成を利用し、まず処理パターンを決定する。その後、その処理パターンに従って詳細処理を取り込む方式をとっている。また、対象分野を事務処理、特にバッチ型ファイル処理に限定し、実用的な事務処理プログラムを合成することを目指している。

2. システムの概要

(1) 仕様書の構成

システムの入力となる仕様書は、ユーザにとって書き易く読み易いものでなければならない。一方システムは実用的な事務処理プログラムを生成できるものでなければならない。そこで、実際に使われている事務処理用プログラムの仕様書を調べ、その特徴を参考にし、仕様書の構成を次のようにした。図1. に仕様書例を示す。

①章立て

四つの章を設け、内容を規定している。

第1章 概要 入出力ファイルと大まかな処理の記述。

第2章 ファイル記述

入出力ファイル、参照ファイル、エラーリストなど、処理に用いるすべてのファイルについての記述。

第3章 処理 プログラムの行う処理の詳細な記述。

第4章 作業領域 特別な作業変数の記述。

②節の設定

処理の記述において、ひとまとめにできる部分は、その部分を表す名前を導入して、後に節を設けてその中で個々を詳しく記述できる。例えば、図1. においては、「区分条件」、「金額計算」という名前を導入し、それぞれ後で設けた、3.1節と3.2節の中で詳しく記述している。

③疑似日本語と疑似コード

基本的に処理は、言い回しに制限を加えた疑似日本語を用いて記述する。しかし、条件や計算等を言葉を用いずに記述できるよう、条件式や代入文が扱える疑似コードを設けている。

1. 概要

取引ファイルを入力し、売上ファイルを作成する。

2. ファイル記述

ファイル名称	FD名	装置	入出力モード	ファイル構成	アクセスモード	レコード長
取引ファイル 売上ファイル	TITORI SIURI	DISC DISC	1 0	SEQ SEQ	SEQ SEQ	34 51

2. 1 取引ファイル

(1) レコードレイアウト

データ項目名	データ表示区分	桁数	データ名
区分	X	(1)	T1-KUBUN
商品コード	X	(4)	T1-SCODE

3. 処理

取引ファイルの各レコードに対して、区分条件を満たす場合、金額計算を行い、売上ファイルに出力する。

3. 1 区分条件

a. 区分 = 1

3. 2 金額計算

a. 単価 * 数量 = 金額

b. 金額 * 0.2 = 割引金額

c. 金額 - 割引金額 -> 値引金額

図1. 仕様書例

(2) システムの構成

システムの構成を図2. に示す。システムは、仕様文解析の前処理として、入力された仕様書より、ファイル名、項目名、節として設けた処理名を取り出し、固有名詞として辞書に登録する。仕様文解析では、文法に基づいて仕様文が解析され、語句がもつ概念とその関係を有向グラフで表わす。仕様獲得では、対応する処理パターンを決定し、そのパターンに処理を埋めていきながら処理内部表現を作成する。その後、その処理内部表現から事務処理用簡易言語を生成する。

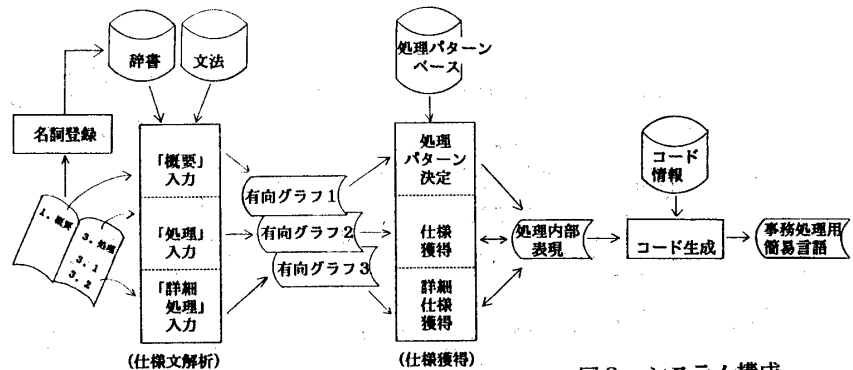


図2. システム構成

A Program Generation System from Japanese Specifications

Takahiko GYOHTOKU, Yumiko ISHIKAWA, Kenji UEHARA

MITSUBISHI ELECTRIC CORPORATION

3. 処理パターンの決定

(1) 処理パターン

システムが対象としている事務処理には、いくつかの定型の処理が存在している。例えば、

- ・トランザクションファイルとマスタファイルのマッチング処理
- ・マスタファイルのメンテナンス処理
- ・トランザクションファイルのチェック処理
- ・レポート作成処理
- ・ファイル作成処理
- ・小計、総計といった集計処理

などである。事務処理プログラムは、こうした処理の組み合わせとして捉えることができる。そこでこれらの処理を処理パターンとして処理パターンベースに貯え、仕様獲得の枠組みとしている。

(2) 仕様文と処理パターン

入力仕様書の第1章の概要では、入出力ファイルと大まかな処理を記述する。これは、システムが仕様書を解析するための指針を与え、詳細な処理の記述を解析するときの場合分けの数を減らし、性能を向上させる効果をもつ。ここで例として、

- 取引ファイルを入力し、売上ファイルを作成する。
- 消費計画入力ファイルを入力し、入力項目検査を行い、消費計画ファイルを作成する。

の二つを考える。aの概要では、ファイルを入力し、ファイルを出力することを述べているのに対し、bの概要では、それに加えてチェック処理を行うことも述べている。これによって、aの処理が一般的なファイル作成処理であるという予測を立てることができるのに対し、bの処理は、ファイルのチェック処理であるという予測を立てることができる。システムはこの予測をもとに、aに対してはファイル入力、ファイル編集、ファイル出力だけに注目して第3章で与えられている処理の詳細を解析すればよい。また、bに対しては、更にチェック処理にも注目して解析することになる。

そこで、第1章概要の仕様文から処理パターンをまず決定し、第3章処理の仕様文からの仕様獲得の効率をはかることにした。

(3) 処理パターン決定知識

入力された概要は仕様文解析部により有向グラフに変換される。その解析結果は、図3.aに示すように、同じ処理について記述したものでも、言い回しの違いによりいろいろな形のものが存在する。また、類似した形をしていても、別の処理を記述しているものもある。こうした概要から処理パターンを決定するには、まず、各処理パターンを入出力ファイルと処理で整理し、他の処理パターンと区別する基準を設ける必要がある。その基準を有向グラフ上の条件として示した例が図3.cである。

すなわち、処理パターンの決定は、概要の解析結果がどの有向グラフ上の条件を満たすか判定することにより、行うことができる。これをシステムでは処理パターン決定ルールとして、

処理パターン名 : 有向グラフ上の条件の形式で記述している。条件の項目としては、

- ・ノードの存在
- ・ノードが持つ属性値
- ・ノード間の関係
- ・ノードの出現順序

がある。

このとき、有向グラフ上の条件は言い回しの違いに依存しないことが望ましい。そこで、解析結果である有向グラフを一旦正規形(図3.b)に変換する。この有向グラフの変換は有向グラフ変換ルールによって行う。

4. 詳細仕様の獲得

入力仕様の概要より処理パターンが決定されたら対応する処理パターンを処理パターンベースより取り出す。処理パターンは、プログラムの制御構造を表わしたもので、その中に具体的な処理を埋め込む箇所が示してある。システムは、その箇所に取り込む内容が第3章処理のどの節に対応するか判断して仕様を取り込む。

5. おわりに

現在、このシステムのプロトタイプを作成中である。今後、システムの効率等を評価していく予定である。

参考文献

- [1] 行徳、他: "日本語ベースの仕様記述に基づくプログラム合成システム"、情報処理第32回全国大会、6G-9。

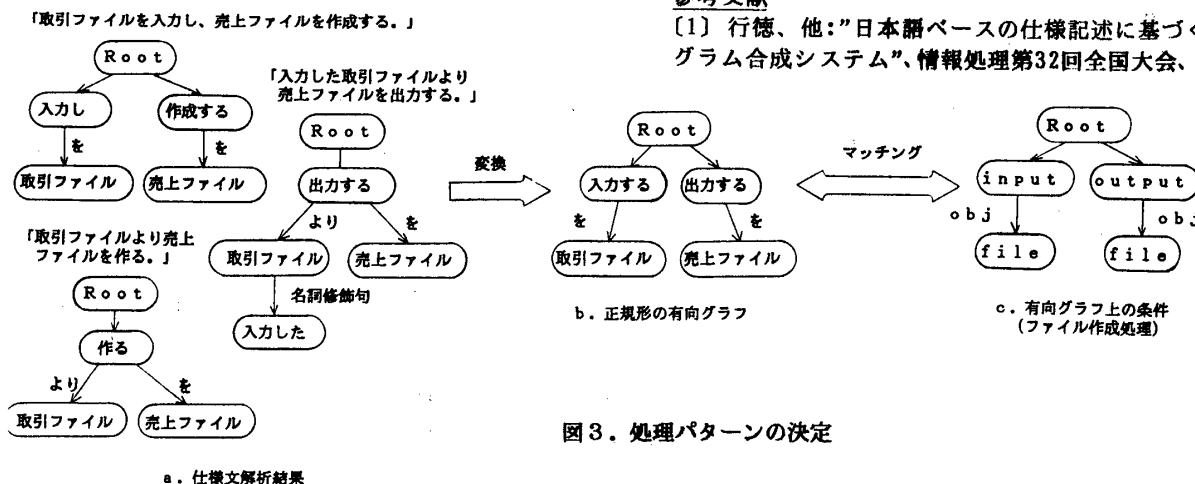


図3. 処理パターンの決定