

プログラム自動生成システムPGENにおける 生成パターンを用いた多言語生成機能

IT-4

杉山高弘^{*}、宮下洋一^{*}、岩元莞二^{**}^{*}日本電気(株) ソフトウェア生産技術開発本部^{**}日本電気ホームエレクトロニクス(株)

1. はじめに

プログラム自動生成システムPGENは、曖昧さや漠然とした人間向きの表現を許した日本文、図、表で記述されたモジュール仕様書から、プログラムを自動生成するシステムである。[1,2,3] プログラム自動生成の分野では、入力仕様書の解析結果である木構造意味表現からプログラム言語やそれを説明する日本文、およびPGENの処理過程を示す説明文等を出力することが必要である。今までのPGENの方式では、それぞれの出力に対応したプログラムを作成していたため、同様な処理を重複して記述しなくてはならなかった。本稿では複数の言語を同一方法で出力できる多言語生成機能を紹介し、複数の言語を生成できる生成パターンの記述方式を提案する。

2. 背景

PGENシステムでは図2.1で示すように自然言語の仕様書を入力として構文解析、意味解析によって木構造のネットワーク表現へと変換する。それらの意味ネットワーク表現から以下を出力している。

- ①構文解析結果の意味ネットワーク表現(Nネット)から解析結果を表わす日本文を出力する。(この出力日本文は入力仕様書に1対1対応する。)
- ②意味解析において曖昧さを一意に決定するために問い合わせ処理を実行する際、ユーザに曖昧な部分を的確に指摘する問い合わせ文を出力する。

③PGENの最終目的であるプログラムを意味解析結果の意味ネットワーク表現(Mネット)から出力する。

④出力したプログラムの処理や、仕様書の省略語をPGENが補完した結果を表示する日本文(レポート)をMネットから生成する。

⑤プログラム構造図またはプログラムスケルトンをPGENのプログラムレポートとして生成することも考えられる。

3. 多言語生成機能の特徴

上記①から⑤の出力は、今まで手続き記述の個々のプログラムで実行されていたが、本稿の生成パターンによる多言語生成機能では、全ての出力方法を同一な書式の生成パターンで記述し、インタプリタが解釈実行する。本機能には主に以下の利点が挙げられる。

(1)仕様書に出現する構文・語彙に依存した言語生成プログラムのブロック構造に対応した節段落や数式混入文を出力しやすい。

(2)ユーザ定義可能な生成パターン

インタプリタにネットワークの探索や手続き呼出しをまかせ、それらの処理から生成パターンを完全に独立させているため、記述量が少なくすむし、記述が容易である。入力となる意味ネットワーク表現は、特定な形に限定されないネットワークや、表現に変更が生じても簡単に処理できる。

(3)可読性の高い生成パターン

出力言語の出力イメージが記述を見ただけで明解である。

(4)手続き的指向を取り入れた高記述性

パターンによる定型な処理では生成できない出力文にも対処できるように、手続き関数を生成パターンの中で呼び出す。これによって、生成パターンを用いない手続き的プログラムに同等な記述能力を持つ。

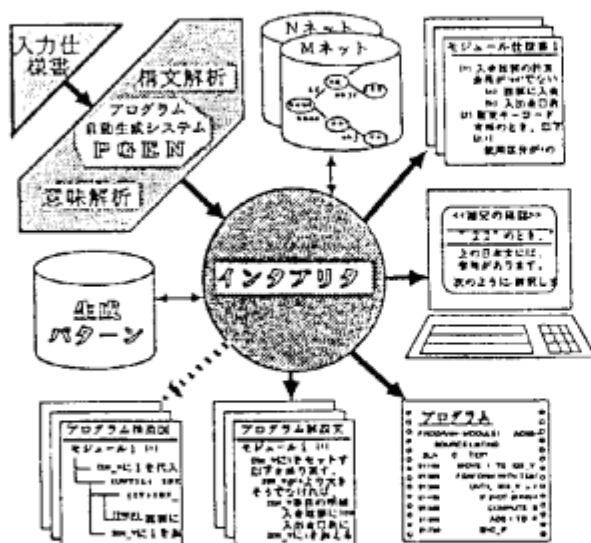


図2.1 多言語生成機能

A Multiple Languages Generator using Patterns in the PGEN Automatic Program Generation System
Takahiro SUGIYAMA, Youichi MIYASHITA, Kanji IWAMOTO
NEC Corporation, NEC Home Electronics, Ltd.

4. 多言語生成機能

特定な形態に制限されない意味ネットワーク表現による概念表現を入力として、出力言語の原形となる生成パターンに従って、そこに表現されている概念を目的の言語(日本文、プログラム、プログラム構造図)によって出力する。生成パターンの記述法とそれを解釈実行するインタプリタについて以下で説明する。

4.1 生成パターン記述

意味ネットワーク表現の各ノードに対応し、多言語を出力する方法を記述した生成パターンを図4.1に示す。変換パターンのシンタックスを図4.2に示す。

図4.1



図4.1 生成パターン例

して取り出し、各構成要素に対して適切な処理を実行する。処理内容を以下の表4.1で示す。

表4.1 生成パターンインタプリタ処理内容

構成要素	処理内容
*スロット名	現ノードから記述されているスロットの子ノードを探索する。
*関数名 アトム	記述されているLisp関数を実行する。アトムに対応する出力文字列(英数字、漢字コード等)を出力文字列テーブルから抽出して出力する。この際、活用語については、分脈によってふさわしい活用形を自動的に抽出する。
[構成要素...]	[]の内部で *スロット名に記述されているスロット以下に子ノードが存在したときのみ [構成要素...] 全体を実行。

インタプリタは、手続き的処理を既存のプログラム言語で記述した関数を実行すること(出統的指向の導入)によりパターンによる定型処理では生成できない出力文にも対応でき、しかも生成パターン記述の簡潔さを失うことがない。

出力文生成時に特に重要なことは、ふさわしい命令語や滑らかな日本語を出力したり、文脈にふさわしい語や活用形や接続詞を自動的に抽出(文脈解析)しなくてはならない。そのために、属性文法を用いることによって現在のノードのまわりの文脈を探索し、文脈情報を伝達

4.2 インタプリタ

インタプリタは、Mネット(Nネット)を入力として現在のノードに対応するパターンを抽出する。現在出力すべき言語が日本語ならばngen識別子以下のパターン記述を、プログラムならばpgen識別子以下を、プログラム構造図ならばsgen識別子以下をそれぞれ取り出す。取り出された生成パターンには、同じ入力に対して状況によって出力の内容を変える複数の生成パターンの候補が存在するため、その中一番ふさわしいパターンを選択する。選択されたパターン構成要素を先頭から分解

する。その文脈情報によってふさわしい語を出力しわけ。この簡単な例として図4.1の中の#period関数がある。現在のIF文が上位IF文の中にネストとして出現するとき、そのネスト情報を生成して現在のノードに渡す。この文脈情報によりピリオドの出力を中止する。

5. 効果

(1) 形式的記述による生成パターン方式を実現したことによって、出力言語のイメージをユーザが簡単に定義でき、入力意味表現構造の変更に対処しやすくなった。これの一つの評価尺度として以下があげられる。

- ・出力言語別にプログラムを開発していたものに比べ、生成パターンとインタプリタを採用したことによって総プログラム数を1/2に縮小することができた。
- ・目的の言語を出力するために、プログラムによる記述が各ノード当り平均7.9ライン必要としたのに対して、本方式では記述量を平均2.9ラインに減少することができた。

(2) 多言語生成の追加機能としてプログラムを解説する注釈文をプログラムの各行に挿入して出力する機能が考えられる。これは、既存のプログラム生成パターンから日本語生成パターンを呼び出すことによって簡単に実現することができ、出力形式の変更にも対処しやすい。

6. まとめ

PGENシステムでは出力言語別にプログラムによる手続き的記述方式で各種文書を生成していたが、本稿で説明したパターンによる形式的記述方式で生成することにした。それによってプログラムの制御部分を出力対象となる言語に依存せず、しかも処理能力を低下させることなく汎用な多言語生成機能にすることができた。現在はノード単位の処理であるが、広域的な出力形態の変化にも応じられるようなシステムに更に発展させていく必要がある。

参考文献

- [1] 西谷、和田、岩元「知識利用のプログラム自動生成システム」情報処理学会第30回全国大会4K-10
- [2] 岩元、西谷、和田、庄司、土田「PGEN-1:仕様管理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
- [3] 和田、西谷、岩元「数式を含む日本語の構文解析」情報処理学会第32回全国大会4S-4

```

生成パターン := (ノード識別名(生成言語名) (生成パターン) 構成要素...)
ノード識別名 := <ノードに対応したN1>または(N1)のノード識別名
生成言語ラベル := pgen | <プログラム生成パターンラベル>
                  ngen | <日本語生成パターンラベル>
                  rgen | <注釈付きプログラム生成パターン>
                  sgen | <プログラム構造図生成パターンラベル>
生成パターン := <同じ入力に対して状況や文脈によって出力の内容を選
                  択するための識別子>
構成要素 := *スロット名 | *関数名 | アトム | [構成要素...]
            <[]には少なくとも一つ *スロット名を含む>

```

図4.2 生成パターンシンタックス