

3L-6

処理パターンに基づいた仕様書理解の評価

鈴木由美子 黒田清隆 土井日輝 行徳孝彦 上原憲二
三菱電機(株) 情報電子研究所

1. はじめに

記述し易さ読み易さといった点を考慮して、自然言語で書かれた仕様を対象にその内容を分析したり、プログラムを生成するシステムが各種研究開発されている。しかし、作成されたシステムの自然言語理解能力には限りがあり、現在のところユーザが期待するほどの柔軟な記述は受入れられない。より良いシステムへ改良する為にも、その質的評価を行うことは重要であるが、評価基準など明確でないのが現状である。

我々は、日本語で書かれた仕様書より事務処理用のバッチ型ファイル処理プログラムを生成するシステムSAGEを開発しているが、今回、その入力仕様の記述の柔軟性に重点を置いてシステムの評価を行った。その結果、適用範囲を限ってはいるが実用的な日本語ベースの入力仕様であるとの感触を得ることができた。

本稿では、その評価方法と結果について報告する。

2. SAGEの概要

2.1 入力仕様書の特徴

SAGEの入力仕様は、図1に示すような章・節構造をもち、記述内容は、操作対象であるデータのファイル、レコード、項目という階層構造のレベルにそって以下のように規定されている。

[1章 概要記述部]: 主にファイルを操作単位とする処理の概要を記述する。

[2章 ファイル記述部]: 使用するファイルの属性と、それを構成するレコードと項目を表形式で定義する。

[3章 処理記述部]: 主にレコードを操作単位とする処理を記述し、項目レベルの操作記述は、節を設けて条件式や数式などを扱える疑似コードで記述する。

[4章 作業変数]: 特別な作業変数を表形式で定義する。

2.2 仕様書理解の概要

SAGEは、まず概要記述文より対象とする仕様書に当てはまる処理パターンの推定を行う。処理パターンは、処理手順の枠組みをモデル化したもので、入力、出力、編集、検査といった処理単位の組み合わせで構成されている。

1. 概要

勤怠データファイルと給与マスタファイルとのマッチング処理を行い、給与明細ファイルを作成する。

2. ファイル記述

ファイル名称	FD名	タイプ	処理モード	編成
給与マスタファイル	KYUUYO	T	I	SEQ
勤怠データファイル	KINTAI	M	I	SEQ
給与明細ファイル	MEISAI	O	O	SEQ

2.1 給与マスタファイルレコード

項目名	識別子名	データ型	桁数
部コード	BCODE	文字	2
氏名	NAME	文字	7

3. 処理

マンナンバをマッチングキーとして、勤怠データファイルと給与マスタファイルに対してマッチング処理を1:1で行い、最後に総数出力処理を行う。マッチング処理ではマッチした場合、給与計算を行い、レコードを給与明細ファイルへ出力する。・・・

3.1 給与計算

もし、残業時間が40以下ならば、
 $(\text{本給}/163) \times \text{残業時間} \times 1.3$ を四捨五入して残業手当とする。
 本給+残業手当→支給額
 それ以外ならば、残業過分処理を行う。
 以上。
 扶養者数×2400→扶養控除

3.1.1 残業過分処理

図1. 仕様書例

次に、推定された処理パターンに基づいて、レコード操作の処理記述文を理解し、その後で、節に記述された疑似コードを解析して、処理パターンの処理単位に取込むことにより、処理の内部表現を生成する。そして、その処理内部表現よりCOBOLベースの簡易言語のソースコードを生成している。

3. 入力仕様の評価

3.1 評価方法

SAGEの処理内容を知らない被験者に、入力仕様の簡単な説明をした後、入力仕様書例を渡し、それを参考にして他の記述による仕様書を複数作成してもらった。作成してもらった各仕様書にはいくつかの評価すべき記述文が含まれているので、そうした記述文を評価項目として挙げ、シ

システムがその記述を理解できたかできなかったか(認識の可/不可)を調べる。今回は、七種類の仕様書を用いて評価した。

3.2 結果

評価項目は103項目となった。その内、レコード操作の記述文(仕様文)をテストした項目が60、節内に記述する疑似コードをテストした項目が43であった。それらの項目の認識結果と修正後の結果を表1に示す。

表1. 処理記述部の評価結果

区分	認識	対応(システムの改良)	個数	修正後
仕様文 60項目	可 (26)		26	可 (51)
	不可 (34)	単語の登録、意味の追加	12	
		文間の関係付け規則の追加	13	
		不適切だとしたもの	9	不可 (9)
疑似 コード 43項目	可 (19)		19	可 (41)
	不可 (24)	構文の追加	22	
		不適切だとしたもの	2	不可 (2)

3.2.1 仕様文記述の評価結果

表1に示すように、認識できなかった記述が34項目あった。その原因を分析し、以下に示す3グループに分けた。(第1のグループ) 単語を辞書に登録する、単語の新たな言回しを追加するといった、単語やルールの追加で対処できる記述である。例えば、「コピーする」という単語や、「1:1のマッチング処理」といった「の」の処理ルールを追加した。

(第2のグループ) 文と文(処理と処理)の関係付けの規則の追加、拡張、修正を必要とする記述である。例えば、前文参照の「それ以外」や、接続助詞「エラーがなければ」の扱いの追加である。こうした文脈に関係した処理は難しい問題を多く含んでいるが、推定された処理パターンに世界を限ることにより、それらの解釈の範囲を狭めることができた。

(第3のグループ) 仕様文として不適切と判断したものである。例えば、「検査を満たす」という記述は、「検査に通る」「条件を満たす」という記述が一般的なため、不適切とした。しかし、以下の4項目は、システムの解析能力不足にも原因があるものである。

①「区分が1であるレコードを入力する」では、「区分が1である」の部分がレコードの項目を対象とした記述であるため、節を設けて記述するという入力仕様の規則に反している。

②「取引ファイルレコードがなくなるまで以下の処理を行

う」では、「以下の処理」が指す処理の範囲を限定できなかった。

③「マッチング処理を行った結果、新売掛金マスタとして出力する」では、「結果」は漠然としたものを指しており、レコードレベル操作の記述に反している。

④2章のファイル記述部にファル名とその属性を定義したからといって、入力ファイルからのレコードの入力を明記していない。

3.2.2 疑似コード記述の評価結果

項目操作の記述は、疑似コードといっても複数の言回しが可能な構文となっている。例えば、

- 金額-単価*0.2->値引額
- 金額-単価*0.2を売上ファイルの値引額とする。
- 値引額に金額-単価*0.2を代入する。

は同じ代入文である。

この疑似コードで認識率は44%と低かったが、構文を追加することにより、仕様文として不適切な記述だと判断した2項目を除きすべて認識できるようになった。

3.3 考察

(1) 今回行った評価の結果、認識率を修正後は全体で89%と高めることができた。記述者の不適切な記述を除けば、さらに高い認識率を持つものと推定される。SAGEの入力仕様書は、章・節構造をもち、記述方法は扱うデータのレベルに従って制限されたものであるが、そうした範囲内であれば、比較的柔軟な記述が可能であるといえることができる。

(2) 認識できなかった記述も、推定されている処理パターンを用いて状態を限定することにより、その意味するところを決定することができた。処理パターンを予め推定する本システムの方法は、仕様文の意味や仕様文間の繋がりの決定を大いに助ける方法であるといえる。

(3) 結果でも明らかなように、記述者は不正確な記述をしやすい。仕様書としての役目を考えれば、記述者には理解できても他の人には理解できない記述は好ましくない。そうした意味でも、不適切な記述を標準的な記述に導くようなシステムの機構が必要である。

4. おわりに

適用範囲は限られるが実用的な日本語ベースの入力仕様であるという感触を得ることができた。今後、この評価結果を参考に、より容易で正確な入力仕様となるよう改良していきたい。

参考文献

- [1] 上原 他: プログラム自動生成システムSAGE, ソフトウェア工学研究会58-10.