4V-6

多国語化を目的とした システム・メッセージの標準化

大平 剛　大場 充
(日本アイ・ビー・エム株式会社 サイエンス・インスティチュート)

1. はじめに

オペレーティング・システムのシステム・メッセージは，現在自然言語で記述されており，ふつう英語である．エンド・ユーザへのメッセージは国語化される例が多くなっている．しかし自然言語によるシステム・メッセージの記述は，曖昧さと使用言語の特性のため，そのままでは機械翻訳にはむかない．1つのメッセージ・ライブラリで複数のユーザ言語をサポートする場合も，計算時間の点で欠点が多い．

上記の問題を解決する方法として，簡単な形式言語を使うことを提案した[1]．今回はその言語で記述されたシステム・メッセージを，実行時にユーザ言語に変換するシステムについて述べる．

2. メッセージ記述言語

システム・メッセージの標準化の目的は，システム・メッセージから曖昧さを取り除き簡潔化することにある．また同時に，多国語へ変換しやすくすることにある．

システム・メッセージが記述すべき意味は基本的に，

1) 状態記述(State_Description)
　システムや対象の状態に関する記述．
2) 対象記述(Object_Description)
　メッセージの記述対象自身に関する記述．

の2つである．そこでメッセージは状態と対象の対の組み合わせで表現できる．

メッセージ言語の構文規則は以下である．

```
<Message> ::= <Message_body> |
              <Message><connective><Message_body>
<Message_body> ::=
      <State_Description>":"<Object_Description>
<connective> ::= "." | "|" |">"
<State_Description> ::=
          <state_predicate> |
          <State_Description><simple_connective>
                              <state_predicate>
<Object_Description> ::=
          <predicate> |
          <Object_Description><simple_connective>
                                  <predicate>
<state_predicate> ::=
          <modal_operator><predicate> |
          "¬"<modal_operator><predicate>
<predicate> ::= <keyword> |
                <keyword>"("<parameter_list>")"
<simple_connective> ::= "." | "|"
<modal_operator> ::= <present> | <past> | ····
```

ここで"."は連言(AかつB)，"|"は選言(AまたはB)，">"は『 ··· なので 〜 をせよ 』なる命令を含む複文を，"¬"は否定を表わす．状態記述部には，時制や可能を示す項<modal_operator>が導入されている．対象記述部には，時制や可能を示す項は必要ない．例えば，

"CP ? : abcdef"

は入力された命令の名前が誤っている場合に出るエラー・メッセージである．これをメッセージ記述言語で記述すると以下になる．

"def pre INV : COMM(CP;abcdef)"

ここで"def"は肯定，"pre"は現在，"INV"，"COMM"はそれぞれ無効(invalid)，命令(command)を意味するキーワードである．

3. メッセージ言語変換

メッセージ変換系が要求されることは，変換系自身が単純で小さく，しかもより自然なユーザ言語に変換できることである．またその際使用される辞書ができるだけ小さいことである．

形式言語で記述されたメッセージを英語や日本語に変換する方法は，基本的には次のようにすればよい．

(日本語)　<object> が <state> である．
(英語)　　<object> is <state>.

ここで<object>,<state>は対象記述，状態記述のキーワードを目的言語へ変換した表現である．

例として，前出の形式言語で表現されたメッセージを考える．

"def pre INV : COMM(CP;abcdef)"　(1)

このメッセージは次の日本語と英語に変換される．このとき"def"は省略され，"pre"は日本語では省略され，英語では"be"の現在形に変換される．

Standardization of system messages for national languages support
Tsuyoshi Ohira, Mitsuru Ohba
Science Institute, IBM Japan, Ltd.

(日本語) 『命令(CP;abcdef)は無効です.』
(英語) 『Command(CP;abcdef) is invalid.』
上の例は単純なメッセージであり，実際にはもうすこし複雑な作業が必要になる．

状態記述に関しては，日本語ではほとんど問題にならないが，英語の場合では動詞と形容詞の区別が必要となる．例えば，

"not pre EXI : DEVI"

ここで"EXI", "DEVI"は存在(exist)，装置(device)を意味する．このメッセージは次のように変換されなければならない．

『装置が存在していない.』
『Device does not exist.』

英語の場合"exist"が動詞であることから，"does"が必要となり，(1)の例では"invalid"が形容詞であることから"is"が必要となる．

命令形が要求されている場合には，日本語では状態記述の語尾変化部"せよ"を追加する．英語では，

<state> <object>

の順に変換する．例えば，

"def res CLE : SCRE"

ここで"res"は責務(responsibility)，"CLE", "SCRE"はクリア(clear)，画面(screen)を意味する．このメッセージは次のように変換される．

『画面をクリアせよ.』
『Clear Screen.』

その他の変換については，以下のとおりである．

連結詞<connective>の処理：

単文のメッセージの間の連結詞をそれぞれの訳語に変換するだけでよい．

否定詞"not"の処理：

英語では状態記述が形容詞なら，"be"の後に"not"を挿入し，動詞なら"do not"を挿入する．

時制<modal_operator>の処理：

日本語では変換しなくても意味が通る．英語では"be", "do"をそれぞれの時制に変換する．

4. メッセージ変換用辞書

状態記述用の辞書は上記で議論したように，形容詞と動詞に分ける．しかし単純にキーワードに対して，原形を登録しただけでは，うまく変換できない場合がある．例えば，

"not pre LOG : USER(xxx)"

を上記の方法で変換する．"LOG"はlog_onを意味するキーワードで，動詞として登録されていれば，以下のようになる．

『使用者名(xxx)が稼動していない』
『User_id(xxx) does not log_on.』

この場合には"log_on"は受身形で変換されなければならない．これを解決するために，"logged_on"を形容詞として辞書に登録すれば，次のように変換される．

『User_id(xxx) is not logged_on.』

進行形も同様に辞書に形容詞として登録する．

例えば，

"not pre REC(MSG) : USER(xxx)"

ここで"REC"は受信(receive)を意味する．辞書には形容詞 receiving と登録すれば変換結果は以下である．

『User_id(xxx) is not receiving(Message).』
『使用者名(xxx)が受信していない(通信文).』

対象記述用辞書は，名詞だけで構成されているので，対応する単語を登録するだけでよい．

5. まとめ

上記のメッセージ変換系を現実のオペレーティング・システムに応用した．実験の対象はＩＢＭのVM/370の核(ＣＰ)である．

ＣＰのメッセージは，1000以上の種類がある．ただ，メッセージのパターン数は，約400種類である．メッセージの生成に必要なキーワードの数は，状態記述に関するものが約200あり，対象記述に関するものは約300ある．

今回の実験でテストしたメッセージは，一般ユーザに対するエラー・メッセージ約100種類である．

"より自然なメッセージ"とは何かを定義することは困難である．したがって変換の評価を数字で表すことはできない．しかし，メッセージの意味が理解できるという意味では十分であると考える．

試行錯誤の過程で，うまく変換できないメッセージが存在した．これは元のメッセージを形式言語に変換する際に，状態と対象をうまく選ばなかったことによるもの(メッセージ自身の問題)である．

またメッセージによっては，補助的に"～の中で"(in ～)，"～によって"(by ～)などを追加した方が自然な場合がある．これは今回の形式言語では表現できない(形式言語の問題)． しかしパラメータとして補助的な対象記述が追加されるので，ユーザに情報を与える役目は十分はたしていると考えられる．

今回の方法では，同じ状態を表す複数のキーワード，つまり受身形と進行形が，同時に辞書の中に存在する場合がある(辞書の問題)．

今後の課題としては，他のシステム・メッセージや言語処理系のエラー・メッセージにこの方法を適用して評価すること．また他のユーザ言語，例えば中国語の変換系を設計することである．

参考文献

[1]大場,大平,『抽象データ型に基づくシステム・メッセージの標準化』,情報処理学会31回全国大会論文集,1985年9月.