

自然言語による仕様記述文解析のための知識の取り扱いについて

田中 克己

(株) 東芝 関西研究所

ソフトウェア等の設計者が仕様書を書く際には、まずひとつひとつの文が明確に記述してあり、仕様書を読む人間により解釈の違いが生じないようにすることが重要である。明確でない仕様記述文には、語彙・構文・意味といったさまざまなレベルでの曖昧性が生じている。現在開発中の仕様記述支援システムにおいては、このような様々なレベルにおける曖昧性・仕様文としての適切さをのチェックを、文解析技術を用いて行なっている。ここでとくに意味解析のための方法として、文の構文解析結果と、対象分野の実体（部品・機器等）の知識との対照により、仕様文の記述の意味的な整合性を検証している。本稿では曖昧性解消のための知識の表現方法ならびに仕様記述文解析の手法・生じた曖昧性と、それを解消するためのユーザに対するメッセージとの関わりについて述べる。

A knowledge-based method for analyzing specifications written by natural language

Katsumi TANAKA

TOSHIBA Corporation, Kansai Research Laboratory

It is important for designers to write non-ambiguous specification document which doesn't have multiple interpretations. We are developing a system which helps a designer to describe specification document using unrestricted natural language and guides him to consistent and non-ambiguous well-formed specification description. Using the knowledge about an objective equipment, the system analyzes the written document from two aspects; the system checks if each sentence follows standard grammar and checks if it specifies meaningful information in the current domain. In this report we explain the knowledge representation, sentence analyzing method, and system-user interaction to reduce ambiguities.

1 はじめに

計算機プログラムの作成に自然言語を利用する試みについては、現在までに様々なアプローチが試みられている [1] が、自然言語で書かれた文章+質問応答によりプログラムを作成していくシステムについての研究が主に行なわれている [2][3]。これは、自然言語で書かれた仕様書に内在する曖昧性・不完全性を、ユーザに対する質問応答という形で解消するものである。このようなシステムは、プログラミング言語に無知なユーザに対してもプログラムの作成を容易にするという点はあるが、実際のソフトウェア作成現場という状況では、このようなプログラム開発過程は想定しにくい。また、実際のソフトウェア作成現場においては、むしろ

- プログラムの要求仕様作成者と、プログラマとの間の仕様のくい違いによる誤動作
- プログラムの再利用（一度作ったプログラムを手直しして利用することが圧倒的に多い）のしやすさ

等が問題となっていることが多い。不完全な仕様+質問応答では、上記の目標を達成することは困難である。そこで今回は、次の立場をとることにする。

- (1) 設計者は自然言語で完全な仕様を記述する。
- (2) システム側ではそのための支援を自然言語処理技術を用いて行なう。

これにより、自然言語でプログラムについての完全な情報が表現され、またプログラムの作成者以外の人にとっても、再利用がしやすいものになる。

このようなアプローチをとる場合には、自然言語による「完全な仕様」の意味が問題となる。すなわち、自然言語ですべての情報を表現した完全な仕様を記述することが最善であるように思われるが、そのために自然言語の持つ簡明さが失われる可能性があるからである。そこで、今回は設計者が実際に仕様記述を行う過程をモデル化し、それを仕様記述言語の書式と、対象知識についての知識として表現した。設計者が記述した文に対しては、定められた書式・知識との整合性を文解析技術を用いてチェックする。本報告では、現在開発中の仕様記述文解析システム [4][5] における、知識の表現方法・それをもちいた解析手法・ユーザに提示するメッセージの関わりについて説明する。

2 仕様記述言語の処理

自然言語で仕様書を記述することにより、人間にとっでは仕様がよく読みやすくなるという利点が生じることになる。しかし自然言語では形式言語と異なり自由な表現が許されているために、読む人によって解釈が異なる曖昧な表現が生じることがある。また、仕様書としては、

記述者により書式・表現が異なるような記述は避けるべきである。

そこで本研究では設計者の仕様作成過程を分析し、仕様記述として適切と思われるモデルを検討した。その結果に基づき自然言語を用いた仕様記述の書式・表現に関する仕様を作成した。以下本章でその点について説明する。

2.1 仕様記述過程のモデル化

今回は特に家電製品（冷蔵庫）の電子制御について、非形式な仕様書と設計者に対する取材を基に仕様記述過程を調査した。その結果、全体的には次のような順で仕様記述を行なっていることがわかった。

- (1) 大まかな処理単位（以下プロセスと呼ぶ）を決める。
- (2) プロセス内での処理を決める。
- (3) プロセス間の関係を決める。

(2) と (3) は設計者により順序が異なることがある。以下各過程について説明する。

2.1.1 プロセスの決定

設計者が制御仕様をわかりやすく記述するために、全体の処理を抽象的なプロセスに分割する過程である。冷蔵庫の例では、図 1 に示すように、F 室温度制御が定常的な処理を行なうためのプロセスとして導入されており、除霜制御、一気冷凍制御等は例外的な処理を行なうためのプロセスである。設計者はこのように全体処理を、それぞれの機能を思い浮かべつつ分割して、プロセスを決定していく。

2.1.2 プロセス内処理の表現

現実の制御対象の動作を定義する過程である。制御系の仕様においては基本的には、制御対象の機器間で生ずる条件と動作の関係を定義したものであるといえる。その中で定義の種類により分類を試みると、次の 4 種の定義型が見られる。

(1) 動作定義型

ある条件における機器の動作を定義する文である。「(条件)ならば(動作)」という表現に置き換えることが可能である。

例) 電源投入時は、コンプレッサの ON 時間の積算が 6.1 時間になると除霜を開始する。

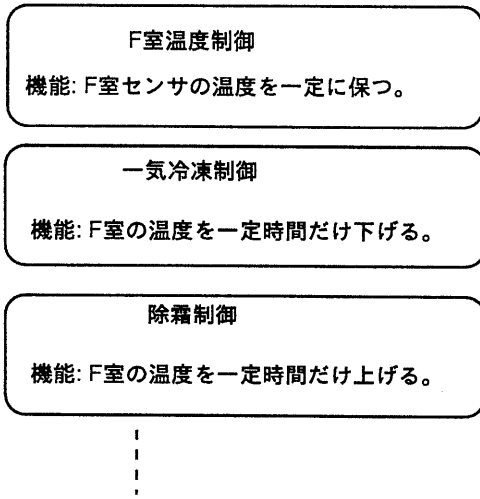


図 1: 導入されたプロセス

(2) 制約定義型

ある一定の条件においての機器に関する制約を定義する文である。「(条件)ならば(制約)でなければならない。」という表現に置き換えることが可能である。

例) コンプレッサーは一度 OFF したら 6 分間は ON しない。

(3) 動作系列定義型

上の (1),(2) の型中で使われる動作を連続した複数の文により定義している表現である。これは動作定義の順序的な連続であるか、または文間を接続詞等で関係づけたひとつの制御構造になっている。

例) 一気冷凍とは、

- 1) 一気冷凍 LED を点灯する。
- 2) 100 分間、コンプレッサーを ON する。
- 3) 一気冷凍 LED を消灯する。

(4) 条件定義型

上の (1),(2) の型中で使われる条件の定義を行っている表現である。

例) 電源投入時、条件 B でモーターダンパーを閉める。

条件 B モーターダンパー開設定温度より低い。

開発者がプログラミング言語により記述する場合には、実際の機器の動作を考慮にいれて記述を行うので、記述はすべて (1) の動作定義型になる。しかし自然言語で記

述する場合には実際の機器の動作を完全に考慮しているわけではなく、また自由な表現が許されるので (2) の制約定義型の表現がみられる。これは実際のプログラミングの際にそのまま表現されるわけではなく、プログラム開発者が手続き的な形に変換して実際のプログラム中に記述している事になる。

(3),(4) については動作・条件についてのマクロ的記述であり、プログラム中ではサブルーチンのような形で記述される事と思われる。

次に条件・動作・状態の定義中で用いられている文の表現については、一般の自然言語に比べるとその基本的な構文は以下のような単純な形に限られている。

(a) (機器) を (動作) する。

例) コンプレッサーを ON する。

(b) (機器) が (状態) である。

例) コンプレッサーが ON である。

ここでいう動作・状態には、上の例で挙げた単純な表現ばかりではなく、その機器の温度、電圧といった属性的な記述を含んだものも多く存在する。

例) ヒーター周辺の温度が 13℃になれば...

また時間を表す副詞句が動詞を修飾している場合が多く存在する。これによって時間的な動作、状態を記述している。

例)

動作: 40 分後、ヒーターを ON する。

状態: コンプレッサーの ON 時間の積算が 10 時間になると...

2.1.3 プロセス間関係の決定

2.1.1 で定義された、プロセス間の関係を定義する過程である。このような関係を定義するためには、設計者が定義したプロセスに対してどのような動作モデルをイメージしているかを知る必要がある。ここで取り上げている、冷蔵庫等の家電製品の制御においては、並列制御対象の規模が比較的小さく、またプロセス間の複雑な関係を実現するのが困難であるという理由から、図 2 に示す階層型のモデルが適当であると思われる。これはプロセス群が階層構造を形成しており、親プロセスが子プロセスを呼び出す形になっているモデルである。

具体的な仕様の記述方法としては、2.1.2 と同様にイベント駆動型の処理を行っており、プロセスを仮想的な機器とみなすことにより、2.1.2 と同様の記述を行なっている。

例) 動作定義型

コンプレッサの ON 時間の積算が 8 時間以上であれば、除霜制御を開始する。一気冷凍制御中に一気冷凍スイッチが OFF であると、一気冷凍スイッチを消灯し、一気冷凍制御を終了する。

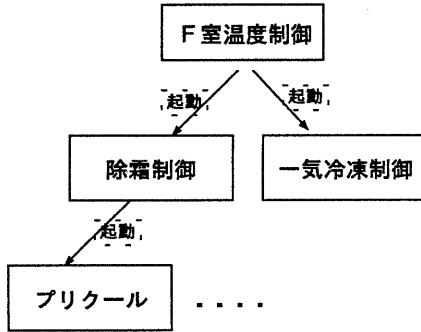


図 2: 階層型のプロセス間関係

例) 制約定義型

一気冷凍スイッチがONであっても、一気冷凍制御を開始しない。

例) 動作系列定義型

一気冷凍スイッチがONであれば、次の1~3を順に行なう。

- 1) プリクールを中断する。
- 2) 一気冷凍制御を開始する。
- 3) 一気冷凍制御が終了したら、プリクールを再開する。

例)に見られるように、現実の制御対象としての機器とプロセスに関する記述が混合している。これからも、プロセスを仮想的な機器とみなし、現実の機器と区別せずに記述する事が自然であるといえる。

また、階層型のモデルにおいては、上位のプロセスで定義された動作、制約等は下位のプロセスでも特に指定のない限りにおいては通用することにすることにより、下位のプロセス内で同様の記述を避けることができる。

2.2 自然言語による仕様記述の形式

自然言語を用いた仕様を設計者が自然に記述するためには、前節で説明したような仕様記述過程のモデルを反映していることが望ましい。そこで、今回は図3に示す仕様記述の形式を設定した。この形式では、構造的には章がプロセスに対応し、各節はそれぞれ構成の定義、条件定義、動作定義、制約定義に対応している。設計者は最上位のプロセスから、章→節の順に記述していけば、仕様記述モデルに即した、自然な表現を書くことができる。この形式にしたがった仕様記述例を図4に示す。

```

<全体仕様> ::= ("プロセス番号" ' , '
               "プロセス" <プロセス仕様> ) *
<プロセス仕様> ::= <構成仕様> <定義仕様>
                  <動作仕様> <制約仕様>
<構成仕様> ::= "プロセス番号" ' , ' 1 ' , ' 構成 '
               (<NULL> | <構成文>)
<定義仕様> ::= "プロセス番号" ' , ' 2 ' , ' 定義 '
               (<NULL> | <定義文> ) *
<動作仕様> ::= "プロセス番号" ' , ' 3 ' , ' 動作 '
               (<NULL> | [ <系列動作> ] |
                [ [局所:] <条件文> ならば <動作文> ] ) *
<制約仕様> ::= "プロセス番号" ' , ' 4 ' , ' 制約 '
               (<NULL> |
                [ [局所:] <条件文> ならば <制約文> ] ) *
<構成文> ::= <プロセス> の ' 下位制御 ' には
               <プロセス> * がある。
<定義文> ::= "条件定義" とは <条件文> である。
               | "動作定義" とは <動作文> である。
<条件文> ::= <機器> が <状態> である。
               | <機器> の <属性> が <状態> である。
               | <プロセス> が <状態> である。
               | <条件文> かつ <条件文>
               | <条件文> または <条件文>
               | <条件文> でない。
<動作文> ::= <機器> を <動作> する。
               | <機器> の <属性> を <動作> する。
               | <プロセス> を <動作> する。
               | <動作文> し、 <動作文>
               | <系列動作>
<系列動作> ::= 次の 1 ~ n を順に行なう。
               ( <文番号> <系列動作文> ) *
<文番号> ::= "数字" ' , '
<系列動作文> ::= <動作文>
               | <時間表現> <動作文>
               | <条件文> ならば <動作文>
               | <条件文> ならば <時間表現> <動作文>
               | <条件文> まで <動作文>
               | <条件文> まで <時間表現> <動作文>
<制約文> ::= <機器> を <動作> しない。
               | <時間表現> <制約文>
<時間表現> ::= <時間> 間
               | <時間> 後
<機器> ::= "機器"
               | <機器> と <機器>
<動作> ::= "動作"
<状態> ::= "状態"
               | <数値>
<属性> ::= "属性"
<数値> ::= "数字" "単位"
               | <数値> "オペレータ" <数値>
<プロセス> ::= "プロセス"

```

図 3: 制御分野仕様記述形式

1. F室温度制御
 1. 1 構成

F室温度制御の下位制御には、除霜制御・一気冷凍制御がある。
 1. 2 定義
 1. 3 動作

局所：コンプレッサは、F室センサの温度がF室指定温度+5℃以上の時、ONする。

局所：コンプレッサは、F室センサの温度がF室指定温度未満の時、OFFする。

一気冷凍LEDがOFFのとき、一気冷凍スイッチがONであると、一気冷凍LEDを点灯する。

一気冷凍LEDがONのとき、一気冷凍スイッチがONであると、一気冷凍LEDを消灯する。

コンプレッサのON時間の積算が10時間以上であれば、除霜制御を開始する。

電源ON直後かつコンプレッサのON時間の積算が4時間以上であれば、除霜制御を開始する。

一気冷凍スイッチがONであれば、一気冷凍制御を開始する。
 1. 4 制約

コンプレッサがOFFであれば、10分間はコンプレッサをONしない。
2. 除霜制御
 2. 1 構成

除霜制御の下位制御には、プリクール・除霜がある。
 2. 2 定義
 2. 3 動作

次の1～5を順に行なう。

 - 1) 除霜LEDを点灯する。
 - 2) プリクールを開始する。
 - 3) 除霜を開始する。
 - 4) ファン遅延を開始する。
 - 5) 除霜LEDを消灯する。
 2. 4 制約
3. プリクール
 3. 1 構成
 3. 2 定義
 3. 3 動作

次の1を行なう。

 - 1) コンプレッサを40分間ONし、Eファンを40分間ONする。

一気冷凍スイッチがONであれば、次の1～3を順に行なう。

 - 1) プリクールを中断する。
 - 2) 一気冷凍制御を開始する。
 - 3) 一気冷凍制御が終了したら、プリクールを再開する。
 3. 4 制約
4. 除霜
 4. 1 構成
 4. 2 定義
 4. 3 動作

次の1～2を順に行なう。

 - 1) コンプレッサをOFFし、EファンをOFFする。
 - 2) 除霜ヒーターをONする。
 4. 4 制約

一気冷凍スイッチがONであっても、一気冷凍制御を開始しない。

2.3 自然言語仕様の処理

2.2で説明した自然言語仕様をもとに、プログラムを作成するための第一段階として、設計者が自然言語により記述した仕様を解析し、その後のプログラム作成過程を考えた形式的表現に変換する必要がある。この処理過程において、解析する項目は次の2点である。

1. 章・節等の形式
2. 節内の文の語彙・構文・意味

処理過程では、これらの表現の解析をそれぞれ行なっていく。その際、仕様記述として不適切である、即ち2.2の自然言語表現の形式に適合していない表現に対しては、記述者に対して指摘して、正しい仕様となるような支援を行なう必要がある。ここで解析・指摘・支援を行なうべき項目と方法を以下説明する。

2.3.1 語彙の解析

図4のような自然言語仕様記述で使用されている語彙には、以下の3つがあげられる。

- (1) 制御分野で用いる機器に関する語彙。

冷蔵庫においては、コンプレッサ、センサ等の機器と、それらの動作・状態の表現。これは予め対象分野ごとに用意することにより、解析が可能である。
- (2) 記述者が定義した語彙。

記述者が定義した、プロセスとそれらの動作・状態に関する表現。これは仕様記述の自然言語表現を解析することにより得られる。
- (3) 通常仕様記述に用いる語彙。

対象分野によらない、一般的な表現。「～の時には」「～の下位制御には」といった表現がそれにあたる。

この3つに対応しない表現は未知語として指摘することになる。その場合には制御分野に関する語彙の追加により対応する。

2.3.2 構文・意味の解析

記述された文に対して、2.3.1の処理を行なった後に、その文が図3の形式中のどの部分に対応するかを、構文・意味解析技術を用いて決定する必要がある。ここで解析すべき項目としては、次の2点があげられる。

図4: 自然言語仕様例

(1) 対象分野に依存した構文・意味関係の解析。

制御対象分野によって、語彙間の構文的関係が決定される場合である。主に動詞を中心とした格関係が解析の対象となる。例えば、冷蔵庫の世界においては、コンプレッサに対しては、状態として {ON,OFF} があるが、その情報を用いることにより、次のような自然言語表現

コンプレッサのON時間が5時間以上ならば、...

に対して、[コンプレッサ]と[ON]、[時間]の関係から、「コンプレッサの状態[ON]の継続時間が5時間以上であれば」の意味であるという解析が可能になる。

このような文を構文・意味解析し、図3の形式との対応をとる。このような解析を行なうためには、対象分野に関する機器とその状態・動作に関する知識が必要になる。

解析に失敗した文については、表現が不適切であるか、対象分野に関する知識が欠落しているかのどちらかが考えられるので、その旨を指摘し、修正するための支援を行なうことになる。

(2) 一般的な構文・意味関係の解析

制御対象分野によらない、大局的な構文的関係である。例えば、

- (条件) ならば (動作)
- (条件) の時 (動作)
- 5分間 (動作)

といった文の構造の解析である。図3に示す自然言語仕様では、このような構文は限定されたものしか許されていない。これ以外の構文が登場した時には、仕様記述文としては不適切であるとみなし、その旨の指摘を行なうことになる。

3 対象知識

2.3.2に述べた意味関係を解析するための知識を活用するためには、Extra-Linguisticな情報が必要になる。そのような意味関係を自然言語から抽出し、それにもとづいた知識・プログラム構造のモデルを生成するアプローチも考えられる [6]。実際には、制御対象の機器等は、設計者にとって常識的な知識といえるので、わざわざ自然言語を用いて書くことはしないと考えられる。そのような前提に立って、それらの機器間関係を記述するために自然言語を用いることが適切と思われる。そこでそれらの基本的な知識を記述するための知識表現の枠組を構成し、冷蔵庫の制御分野における対象知識を作成した。本研究では、対象知識の役割を以下の2点とした。

(1) 2.3.2に述べたような、語彙間の意味関係を解析するための対象世界の表現

(2) 変換の結果としての形式的表現を生成するための枠組

(2)の役割を加えた理由は、次の2点である。

- 対象世界の知識は静的ではない。記述者が仮想的な機器としてのプロセスを導入したり、時間的表現を用いたりした時に、対象世界のモデルは変化するので、対象知識を操作する手段が必要となる。
- 対象知識に基づいた解析を行なうので、生成される表現は対象知識に自然言語仕様から得られた知識を付加した形式を用いた方が、データ形式が統合されているのでその後の仕様検証段階にとって都合が良い。

この役割を満たすため、対象知識は図5に示すような意味ネットワーク構造にした。基本的にはノードとノードの間を有向リンクで結ぶ構造である。ノード名・リンク名については特に制限はない。また言語解析用辞書とノード間の対応をとるために、辞書の見出し語・品詞の組とノードの対応関係を示すデータが存在している。

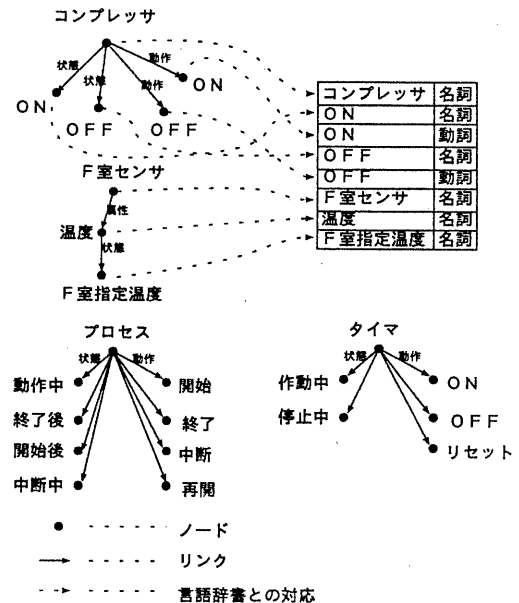


図5: 対象知識の構造

変換処理を行なう前の対象知識は、図5のように、機器ノードとその状態・動作リンクで結ばれた状態ノード・動作ノードが存在するだけである。ここで仮想的な

機器として、プロセスと時間的表現を表すタイマに関する知識を今回は導入した。

この意味ネットワークに対しては設計者が簡単に以下の操作を行なうことが可能である。

- 新たなノードの生成
- 新たなリンクの生成
- ノードと解析用辞書（見出し語・品詞の組）との対応関係データの作成
- 新たなインスタンスの作成

ここで言うインスタンスとは、図6に示すように、新たなプロセス・時間的表現が設計者により、仕様記述中で導入された場合、新たなノードとその状態・動作を生成するためのものである。これにより、対象世界のモデルの操作が容易になる。

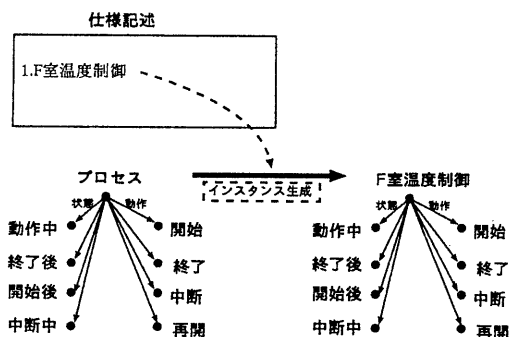


図6: 新たなインスタンスの生成

解析・変換処理中でこのような操作を繰り返すことにより、操作を加えられた結果の意味ネットワークが、最終的な形式的表現となる。

上記の枠組により、設計者による知識の操作を最小限にし、かつ記述した自然言語文の妥当性の検証が可能になる。

4 仕様記述解析システム

今回は自然言語による仕様を第2章で定義した表現形式に基づいて解析し、形式的表現に変換するための実験システムを試作した。この実験システムでは冷蔵庫の制御仕様について、表現が適切である場合は対応する形式的表現を作成する。また不適切な表現についてはエラーメッセージを出力し、文の修正を促す。本章では実験システムの構成、解析のメカニズム、結果とそれに対する考察を説明し、実用化するために必要な課題を検討する。

4.1 システム構成

今回作成したシステムの構成を図7に示す。このシステムを用いた場合の典型的な処理の流れを説明する。

ユーザが仕様記述支援エディタを用いて書いた自然言語仕様記述中の文に対して、仕様記述文解析処理を行ない、仕様記述文としての適切さをチェックする。適切な入力文は形式的な構造に変換する。不適切な入力文に対しては解析に失敗し、エラーメッセージを出力する。適切な入力文に対しては、解析に成功し、ユーザへの提示を行なうための形式的表現を作成し、出力する。出力結果は仕様記述エディタ上でユーザに対し提示される。この処理は入力文全体に対して一括的に、あるいは入力文一文ごとに行なうことができる。

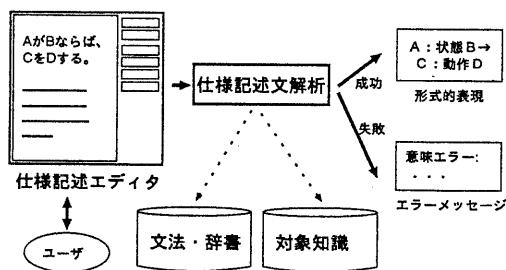


図7: 実験用変換システムの構成

4.2 仕様記述文に対する解析処理

本節では仕様記述文に対する解析処理の過程を各段階ごとに例文を用いて説明する。言語解析部においては、図8に示す手順で処理が行われていく。

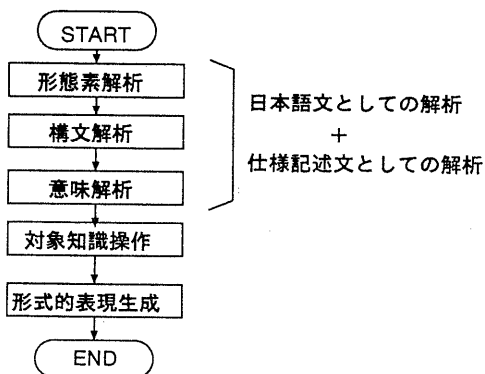


図8: 仕様記述文解析の過程

形態素解析・構文解析・意味解析の各段階においては、通常の日本語文としての適切さと、仕様記述文としての適切さの両方を考慮した解析が行なわれる。通常の日本語文の解析は、対象知識で用いる語彙の他に、日本語としての一般的な語彙を辞書に含んでおり、日本語文として解析が行なわれた後に、語彙・構文・意味を対象知識を参照することによりチェックすることにより、仕様記述文として適切といえるかどうかの解析を行なっている。意味解析段階において例を示すと、日本語文としての意味解析結果は、図9のようにになっているが、対象知識を参照することにより、語彙間の意味関係に基づき、図3の形式との対応づけを行なうことになる。図9の場合には、意味解析結果における格関係の結び付きと対象知識の整合がとれているので、解析に成功したことになる。

このように解析のフェーズを2段階にわたり設けている理由は、ユーザ・インタフェースをより適切に、きめ細かくすることにある。ユーザは、自分としては通常使いなれた、日本語を入力しているので、解析に失敗した時にどのように対応してよいかわからなくなることが予想される。ここで解析に失敗した時に、その原因が、

- 日本語としては、システムでは理解不能な文を入力した。
- 仕様記述文として、意味不明な文を入力した。

ものであるか、ということシステム側で判断することができれば、ユーザへの対応として、つぎのうちのいずれを示唆するかを決定することができる。

- 仕様記述文の書き換え（日本語として解析可能な文にするための）
- 仕様記述文の書き換え（仕様記述文として意味をもつための）
- 対象知識（仕様記述文としての意味をもつかどうかの判断基準）の操作

解析処理の後には、解析結果の構造を基に対象知識を操作する（図10）。ここでは、機器の状態ノードから機器の動作ノードへ、動作を促す message リンクを新たに張っている。

最後に形式的構造から形式的表現を生成する。

各段階において、解析に失敗した場合にはその部分を指摘するエラーメッセージを出力する。単語の解析エラー（図11）、意味解析エラー（図12）の例をそれぞれ示す。

仕様記述文:
コンプレッサは、F室センサの温度がF室指定温度未満の時、OFFする。

意味解析結果(日本語としての):

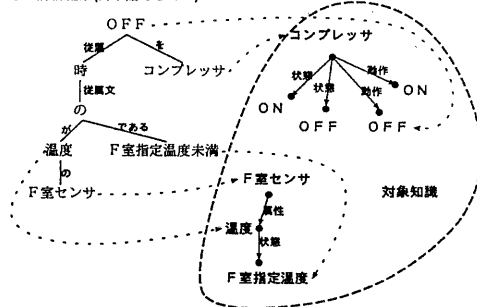


図9: 意味解析時における対象知識との対応づけ

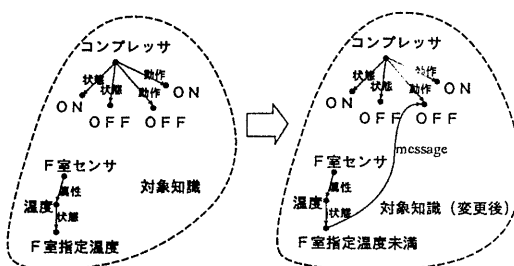


図10: 対象知識の操作

4.3 今後の課題

(1) 支援機能の強化

今回作成した自然言語仕様記述では、構文にある程度の制限を加え、解析に失敗した文に対してはエラーメッセージを出力することになっている。しかし、設計者が自由に自然言語により仕様を記述した場合には、仕様記述文として不適切であったり、曖昧性があったりする文が数多く存在するものと思われる。今回はそのような場合には、図11、図12のようなメッセージを設計者に対して提示することにより対処を行なっているが、これだけでは設計者が何を変更して良いか不明確な場合がある。例えば、図12の例でも、仕様記述文中のどの部分を修正したら良いかが、エラーメッセージだけでは明確ではない。[一気冷凍LED]に対して、とりうる動作（点灯または消灯）をユーザに対して提示することがより親切な支援であるといえよう。

このような場合には、例えば曖昧性が生じた場合には複数の候補を提示し、選択させることにより

仕様記述文: F室センサの温度が3℃以上ならば、コンプレッサを運転する。

エラーメッセージ: [運転]が対象知識にありません。

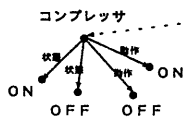


図 11: 単語解析エラーの例

仕様記述文: 一気冷凍スイッチがONであると、一気冷凍LEDをONする。

エラーメッセージ: [-一気冷凍LED]と[ON]の関係が対象知識にありません。

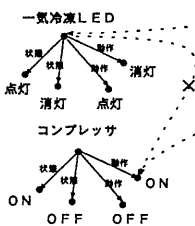


図 12: 意味解析エラーの例

修正が簡単に行なえるようにする必要がある。自然言語処理技術(特に生成技術)と知識処理を用いることにより、将来的にはよりきめ細かい支援を行なう必要がある。

(2) 一般化されたモデルに対する適用

今回作成した仕様記述のための自然言語仕様は、家電製品のような小規模な機器の制御に対する仕様記述過程のモデルに基づいて作成されたものである。制御対象分野が異なった場合には、ここで論じたモデルでは不十分な点も考えられ、それは自然言語の書き方にも影響を与える。自然言語を用いた仕様記述を一般的なモデルに対して適用するためには、このような仕様記述過程を一般化したモデルにしたがった、自然言語仕様の設定方法を検討していく必要がある。

5 おわりに

このような問題では、自然言語記述・知識の両方をよりユーザにとって自然で、記述量が少なく済むようにするのが最善である。今後は、プログラムの作成を含めた実際の利用を通して、自然言語記述・知識の妥当性を検証していきたい。

参考文献

- [1] 辻井 潤一, 上原 邦明. ソフトウェア工学と自然言語処理. 情報処理, Vol. 28, No. 7, pp. 913-921, 1987.
- [2] 杉山 健司, 秋山 幸司, 亀田 雅之, 牧之内 顕文. 対話型自然言語プログラミングシステムの試作. 電子通信学会論文誌, Vol. J67, No. 3, pp. 297-304, 1984.
- [3] 大西 淳, 阿草 清滋, 大野 豊. 要求定義のための要求フレーム. 情報処理学会論文誌, Vol. 28, No. 4, pp. 367-375, 1987.
- [4] 田中 克己, 他. 自然言語を用いた仕様記述支援システム - 仕様解析部 -. 情処 43 回全国大会, pp. 5-407, 1991.
- [5] 野村 浩一, 他. 自然言語を用いた仕様記述支援システム - インタフェース部 -. 情処 43 回全国大会, pp. 5-409, 1991.
- [6] 佐伯 元司, 蓬来 尚幸, 榎本 肇. 自然言語仕様からモジュール構造を得る手法について. 情報処理学会論文誌, Vol. 30, No. 11, pp. 1479-1493, 1989.