

自然言語仕様書に基づくソフトウェア開発支援システム

4Q-2

— システム構成とプログラム合成手法 —

田中克己

池田朋男

出口豊

(株) 東芝 関西研究所

1 はじめに

ソフトウェア開発の再上流工程においては自然言語仕様書が用いられるのが一般的であるが、自然言語仕様書をその後の工程に利用することは困難である。その最大の原因としては、次の2点が指摘されている。

- 自然言語文の持つ曖昧さ (意味の多義性)
- 自然言語文書の持つ断片性 (文間の関係の不明確さ)

筆者らはマイコン制御用プログラムを対象として、自然言語仕様書を再上流としたソフトウェア開発支援システムを開発している。ここでは、文の曖昧性については、対象分野で用いられる文表現のモデルを定義し、仕様文を解析することによりモデルに基づいた文の曖昧性・適切さの判定を行ない、必要に応じて修正のためのガイドを行なうことにより、モデルに適合した仕様書の作成を支援するシステムの研究を行なっている [1]。自然言語文書の持つ断片性 (文が断片的に記述されており、その間の関係が明示されていない) については、解析した文を実行可能表現に変換する際の問題となる。今回、制御分野で広く用いられている、状態遷移表現に基づく実行可能仕様 [2] を生成することを目的として、プログラム自動生成のための基本方式を提案する。

2 システム構成

図1にソフトウェア開発支援システムの構成を示す。設計者は、仕様記述エディタを用いて仕様書を記述し、システム側は仕様書内の一文一文を、辞書・文法・対象知識により表現された仕様記述文のモデルに従い解析する。その結果、曖昧性を持つか、あるいはモデルに適合しない文についてユーザに指摘し、ガイダンスを提示することにより対話を行なう。その過程を繰り返すことにより最終的にモデルに適合した文から成る仕様書が作成され、文は対応する形式表現に変換される。その後、形式表現より状態遷移表現の基本的要素である状態・遷移アーク (ある状態において、イベントが発生した時のアクションと次に移る状態を決定する関数) を決定することにより、状態遷移表現を作成する。

3 自然言語仕様文のモデル

制御分野に登場する概念として、以下の要素を定義する。

状態 各機器のとりうる状態 (例. コンプレッサ -ON,OFF)

A Software Development System Based on Natural Language Specifications
TANAKA Katumi, IKEDA Tomoo and DEGUCHI Yutaka,
TOSHIBA Kansai Research Laboratory

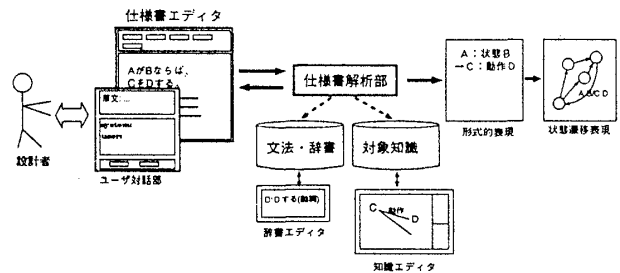


図1: ソフトウェア開発支援システムの構成

イベント 状態の自然な変化 (例. センサ温度 - 0℃ → -10℃)
 アクション 状態の能動的な変化 (例. コンプレッサ - OFF → ON)
 条件 各機器がある状態にあるかどうかの判定、条件の論理的結合 (∧, ∨) も条件とする。 (例. センサ温度 < 0℃)
 自然言語仕様文は上記の要素間の関係を表現したものと考えられ、現在仕様文として基本的に以下の3種類の表現が有効である。

- (1) 動作記述
 イベント (+条件) ならば、アクションを行なう。
 例. センサの温度がF室指定温度 + 3℃以上になれば、コンプレッサをONする。
- (2) 制約記述
 イベント (+条件) ならば、アクションを (一定時間) 行なわない。
 例. コンプレッサがOFFになれば、コンプレッサを6分間ONしない。
- (3) 系列動作
 順的に行なう処理を記述する。アクションのみの記述と、イベントまたは一定時間が経過するまでの (待ち) の記述が並んでいる。 例. 次の1~3を順に行なう。
 1) コンプレッサをONする。
 2) 3分間待つ。
 3) コンプレッサをOFFする。

4 自然言語文と状態遷移表現とのギャップ

今回扱う仕様書においては、自然言語仕様文と状態遷移表現間のギャップは主に次の2点である。

- (1) 具体的なイベント、アクションを自然言語文が直接表現していない。抽象的な記述 (例えば時間表現) を状態遷移表現で直接記述することができない。

(2) 自然言語文は前章で説明した動作記述・制約記述が混在して記述されており、状態遷移表現における「状態」のように相互を結び付ける要素がない。

ここで、(1)については、状態遷移表現内で扱うことが可能な、具体的な機器(タイマ等)を導入し、形式表現を変換することによる解決を試みた。また、(2)については、設計者の制御分野における仕様記述過程について、以下のように考えた。

- (a) ある状況を想定し、
- (b) その状況で起こる得るイベントを想定し、
- (c) イベントが発生した後の(アクションによる)状況の変化のしかたについて考える。

ここで状況の集まりを効率的にまとめたものを状態遷移表現におけるシステム状態として設定している。自然言語仕様書には、そのように、設計者が抱えている陽に表現されない「状況」が記述されているものといえる。

上記の仮定に基づき、断片的に記述されていて関連が明示されていない文間は、各文の持つ状況を通じて以下のように関連づけることにした。

- (1) 自然言語文のそれぞれは、システム内の状況間の関係について必要十分な意味を持つ。
- (2) 仕様書における同一節内で記述された自然言語文の集まりは位置に関係なく、連言的な関係を持つ。すなわち文はそれにより記述された状況内では必ず成立する。

5 状態遷移表現への変換手法

前章の方針に基づき、状態遷移表現への変換を以下の過程で行なう。

- (1) 陽に表現されない機器の導入
時間表現等について、システム内で対応する機器(タイマ等)を導入し、形式表現を変換する。
- (2) 制約記述から動作記述への変換
制約記述は動作記述中のアクションに対する制約であると考えられるので、動作記述に対してイベント・条件を付加するように変換を行なう。
- (3) 状態の作成
状態遷移表現は、システム内で考えられるすべての状況を記述したものであると考えられる。それは機器の状態の直積により与えられる。
- (4) 状態間の遷移アークの作成
動作記述を用いて、仕様に適合する遷移アークを作成する。これは、各状態より可能なイベントを発生させ、各動作記述の条件・イベント部とマッチングをとり、マッチした動作記述部のアクションを実行することにより行なわれる。その際イベント・アクションによりシステムの状況が変化するので、それに対応した状態を検索する。これが遷移アークの行き先に相当する。
- (5) 遷移に関連しなかった状態を削除する。
冷蔵庫制御についての簡単な変換例を図2に示す。

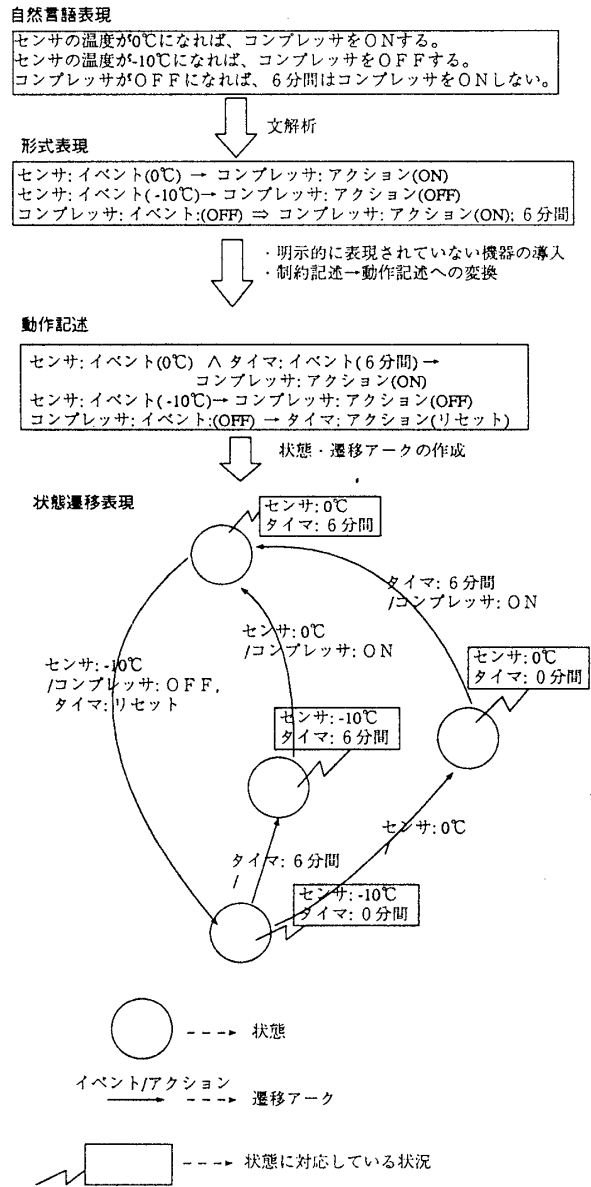


図 2: 状態遷移表現への変換過程

6 おわりに

制御分野について、記述モデルに適合した自然言語仕様書より、状態遷移表現を作成する手法について述べた。今後は、自然言語仕様書におけるより多様な状況間の関係を表現するモデルに拡張するとともに、仕様の正当性・完全性の検証を含めた仕様書作成支援方式について検討を進める。

参考文献

- [1] K.Tanaka et al. Computer-assisted Writing System for Specification Documents Described in Natural Language. *Proc of NLPRS '93*, pp. 76-84, 1993.
- [2] 岸 他. 状態遷移モデルに基づくプログラム部品合成システムの開発. 情報処理学会研究会報告, Vol. 80, No. 22,, 1991.