

自然言語仕様書からのソフトウェア自動作成システムの開発

4C-2

出口豊[†] 田中克己[†] 池田朋男[†] 岸本卓也^{††}
 (株)東芝 関西研究所[†] 東芝 AVE(株) HA 開発事業所^{††}

1 はじめに

ソフトウェア開発における CASE ツールとして、状態遷移図を利用し仕様を記述するシステムが多く提案され実用化されている [1]。状態遷移図を用いて仕様を記述するシステムには、状態遷移図の記法に慣れないと理解・記述しにくく修正にも時間がかかるという問題点や、部品化しづらいため状態遷移図の再利用が難しいという問題点がある。そこで、我々は自然言語仕様書より状態遷移図を自動生成し、生成した状態遷移図を用いてソフトウェアを作成するシステムを提案し、研究開発を行っている。自然言語で記述された仕様書は、状態遷移図と比較して、多くの人が容易に理解でき、保守や再利用が容易に行えるという利点を持っている。また、自然言語仕様書より状態遷移図を作成することで、仕様書とソフトウェアの一致性を保つことができる点も大きな利点である。今回、既に製品化されている冷蔵庫を対象に、自然言語仕様書を作成し、本システムを用いてソフトウェアを自動生成する実験を行った。ソフトウェア生成の途中の段階で生成される状態遷移図に関しては、仕様を直接記述した従来の状態遷移図とほぼ同等の大きさのものを自動生成することができた。最終的に生成されたソフトウェアは、実機上で正常に動作した。ここでは本システムの概要及びソフトウェア自動生成の実験結果について報告する。

2 ソフトウェア作成処理の流れ

図 1 に本システムにおけるソフトウェア作成処理の流れを示す。図 1 中の線で囲んだ部分 (状態遷移図変換以降の部分) は、既に IDSS- μ [1] において実現及び実用化されている部分である。図中の各段階について以下に説明する。

仕様記述

設計者は、専用の構造化エディタ (仕様書エディタ)[3] を用いて、システム側で規定されている文法にそって仕様書を記述する。冷蔵庫制御の機器の情報などの、設計者にとって常識的な知識に関しては、専用の知識エディタを用いて木構造的に記述する [2]。

設計者が仕様を記述する際には、まず大体の処理単位 (以下プロセスと呼ぶ) を決定し、その後プロセス内の処理とプロセス間の関係を決定している [2]。本システムでは、複数のプロセスが並列に動作する場合には各

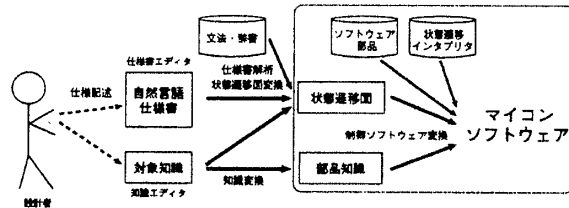


図 1: 自然言語仕様書からのソフトウェア自動生成処理

プロセスごとに仕様書を作成する。1つのプロセスに対しては1つの章の形で記述し、プロセス間の関係については上位プロセスが下位プロセスを呼び出す階層型の記述を行う。

仕様書解析

仕様書内の一文一文を、規定された文法や辞書・対象知識によって表現された仕様記述文モデルに従い解析する。曖昧性のある文やモデルに適合しない文に関しては、システムよりエラー情報を表示し設計者に指摘する。最終的にモデルに適合する文のみからなる仕様書が作成される。

状態遷移図変換・知識変換

仕様書内の文は、ある条件のもとであるイベントが起こった時に行う動作について記述されている。仕様書解析結果を状態遷移図に変換する際には、最初に、各動作に関して、実際に動作が発生するための必要十分条件を仕様書解析結果より求めておく。状態遷移図は、各状態においてイベントが発生した際に動作が起こるか否かを前段階で求めた必要十分条件を用いて判断し、動作が発生した場合には新たな遷移アークと (必要であれば) 新たな状態を作成するという操作を、初期状態より順次行うことで作成される [4]。最終的に、プロセス単位ごとに章を作成して書かれた仕様書に対応した形で、階層化された状態遷移図が仕様書ごとに作成される。

仕様書中にみられる時間的表現 (例.6分間 ON しない) などの、直接実装が不可能なオブジェクトに関しては、実装を意識したオブジェクト (タイマなど) を導入することで必要十分条件の形で表現することを可能にしている。

また、仕様記述時に作成された知識のフォーマット変換も同時に行う。

制御ソフトウェア変換

状態遷移図からのソフトウェア生成は、以前に当社で開発された IDSS- μ [1] を用いて行う。IDSS- μ では、あらかじめアセンブラで開発された状態遷移インタプリタを用い、状態遷移図より変換された状態遷移テーブル

Software Synthesis System from Natural Language Specifications
 DEGUCHI Yutaka[†], TANAKA Katsumi[†], IKEDA Tomoo[†]
 and KISHIMOTO Takuya^{††}

TOSHIBA Kansai Research Laboratory[†], TOSHIBA AVE HA
 Operations Center^{††}

E-mail: {deguchi,tanaka,tommy}@krl.toshiba.co.jp,
 kishi@jlab.toshiba.co.jp

及び部品知識を参照しインタプリタ方式で動作する制御ソフトウェアを実現している。制御ソフトウェアへの変換は、状態遷移図から状態遷移テーブルへの変換及び部品知識の変換を行うことに相当している。なお、この方式で生成されたソフトウェアの性能は、従来のアセンブラプログラムと比較しても大差はなく、既に当社の家電製品において実用化されている。

3 実験

3.1 実験方法

今回、既に製品化されている冷蔵庫の自然言語仕様書を作成し、本システムを用いてソフトウェアを自動生成する実験を行った。生成されたソフトウェアは、実際の冷蔵庫に組み込んでテストを行い、正常に動作することを確認した。本実験では、冷蔵庫の仕様及び本システムに関して熟知したエンジニア1名が、約40時間かけて仕様書を記述した。自然言語仕様書は、全16枚、約600行で記述されている。AS4080上で実験を行った。

3.2 実験結果

従来の状態遷移図及び作成されたソフトウェアと、今回の自然言語仕様書より生成された状態遷移図及びソフトウェアに関して、状態遷移図やメモリ容量のサイズなどを比較した結果を表1に示す。状態遷移図に関しては、従来の状態遷移図と比較して状態数が2.5割ほど増えており、付随する形でイベント数も増えている。プロセス数は従来のものの4倍になっている。プログラムの大きさに関しては、全体では従来と比較して約800バイト減少しているが、インタプリタ部・部品における減少分約1600バイトは使用マイコンを8ビットに変更したために生じた減少分なので、実質的には、本システムの使用により、プログラムサイズは状態遷移テーブル部の増加分約800バイトほど増加している。この程度の増加は実用上大きな問題とはならない。この結果より、少なくとも本実験に関しては、従来とほぼ同等の大きさで実用上問題ないソフトウェアが自動作成できることが判明した。

変換時間に関しては、自然言語仕様書から状態遷移図への変換時間全体のうち、約9割は状態遷移図作成に費やされている。

3.3 考察

本手法では、自然言語仕様書の理解や修正、再利用を容易にするため、プロセスの関係を階層化して記述している。そのため、本手法で生成される状態遷移図は、自然言語仕様書の階層構造を反映した形で階層的に作成され、従来の人手でフラットな(階層構造のない)形で作成された状態遷移図と比較すると状態数が2.5割ほど増加し、プロセス数が4倍増加している。

本システムにおいてソフトウェアの大きさを決める状態遷移テーブルの大きさは、状態遷移図の状態数・イベント数・アクション数に依存するので、最終的なソフトウェアのサイズは、状態遷移図の大きさが増大したことにより約800バイト増加している。また、ソフトウェアの実行速度は状態遷移図のプロセス数に比例するので、本システムで生成されたソフトウェアの実行速度は、状態遷移図が階層化されたことより従来のものの4

表1: 状態遷移図及びソフトウェアのサイズ

		従来	本手法
使用マイコン		4ビット	8ビット
プログラムサイズ(全体)		11,032	10,211
(状態遷移テーブル部)		6,009	6,791
(インタプリタ・部品部)		5,023	3,420
状態 遷移図	プロセス数	16	64
	状態数	204	249
	遷移数	617	457
	イベント数	220	253
	アクション数	186	179

分の1になっている。しかし、このソフトウェアの大きさの増大・実行速度の低下も、実用に関しては特に問題とはなっていない。

ソフトウェアへの変換時間に関しては、状態遷移図への変換時間が約9割を占めている。本実験では、仕様書解析・状態遷移図変換・制御ソフトウェア変換を順に処理している。実用する際には解析時及び変換時に検出された自然言語仕様書のエラーの修正を行う必要があるため、仕様書修正後の解析・変換時に修正した部分のみを部分的に処理する機能を開発し、エラー修正を含めた全体の開発時間を短縮することが必要であると思われる。

4 おわりに

自然言語からソフトウェアを自動的に作成するシステムを開発し、冷蔵庫の自然言語仕様書よりソフトウェアを自動生成する実験を行った。自動生成されたソフトウェアは大きさ・実行速度とも実用上問題なく、実機上でのテストでも正常に動作することを確認した。今後は、状態遷移図への変換時間を短縮する方法や部分的な状態遷移図変換の開発方法を検討するとともに、誤り修正に関する本システムの評価を行っていきたい。

参考文献

- [1] 平山雅之. 小規模ソフトウェア開発支援技術. パソコンリテラシ第20巻第5号, pp28-32, 1995.
- [2] 田中克己, 野村浩一, 武田公人, 平川秀樹, 天野真家. 自然言語を用いた仕様記述システム - 仕様解析部 -. 情報処理学会第43回全国大会論文集(3), pp. 407-408, 1991.
- [3] T.Ikeda, K.Tanaka, T.Chino, and Y.Deguchi. An Editor for Specifications in a Software Development Support System. Proceedings of HCI International 95, pp. 659-664, 1995.
- [4] 田中克己, 池田朋男, 出口豊. 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 - システム構成とプログラム合成手法 -. 情報処理学会第48回全国大会論文集(3), pp.91-92, 1994.