6A-5

UML ステートマシン図を用いた S/W 設計検証方法の提案

炭崎　竜平†　　辰巳　尚吾†　　井上　勝行†　　松村　和機‡　　吉田　実†

三菱電機(株)先端技術総合研究所†　　京都大学大学院情報学研究科‡

１．はじめに

S/W(ソフトウェア)開発の上流工程で早期に不具合を発見するための手法として、設計検証が注目されている。設計検証には、SMV[1]や SPIN[2]などのモデル検査ツールを用いる試みが広がっている[3]。

これらのツールは状態遷移モデルを入力とするので、設計検証を行うためには、S/W 設計資料の中の状態遷移モデルをツールへの入力とすることが望ましい。しかしながら、状態遷移モデルを描くことが困難であるため、設計資料にそれらが完備されていないことが現実には多いこと、およびモデル検査ツールの入力となる状態遷移モデルの記法はツールごとに独自のものであり、その記法は近年主流となっている UML(Unified Modeling Language)[4]ステートマシン図とは大きく異なることが問題となる。これらの問題点がモデル検査ツールによる設計検証を S/W 開発に実適用する際の障壁となっている。

本稿では、これらの問題点を解決するための方法を提案する。

２．S/W 設計検証の流れ

本稿で提案する S/W 設計検証方法の流れを図１に示す。

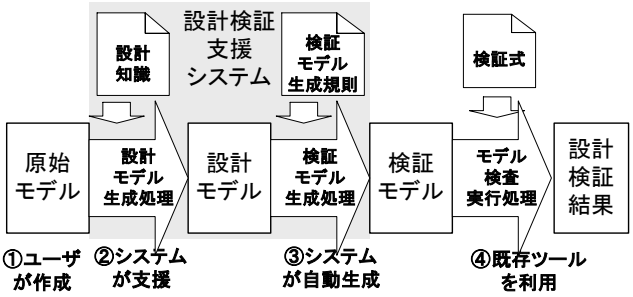

図１　S/W 設計検証の流れ

提案方法を適用した設計検証支援システム(以下、システム)では、設計モデル生成処理機能と検証モデル生成処理機能をユーザに提供する。このシステムのユーザは S/W 設計者である。

①ユーザは、設計資料の内容に従って原始モデルを作成する。原始モデルとは、検証の対象となる UML ステートマシン図の種となるものである。

②ユーザは、システムの支援を受け設計知識を原始モデルに適用し、設計モデルを生成する。設計モデルとは、設計検証の対象となる UML ステートマシン図である。また、設計知識とは、設計モデル生成に必要な対象ドメインに関する知識である。

③システムは、設計モデルに対して検証モデル生成規則(以下、生成規則)を適用し、検証モデルを自動生成する。検証モデルとは、モデル検査ツールへ入力可能な形式の状態遷移モデルである。一方、生成規則とは、UML ステートマシン図からモデル検査ツールの入力形式への変換規則を定義したものである。

④ユーザは、システムが生成した検証モデルを別途作成した検証式とともにモデル検査ツールに入力する。これにより、検証式に記述された特性に関する検証結果を得ることができる。

The proposal of the S/W design verification method using the UML state-machine-diagram

†Ryuhei Sumisaki, Shogo Tatsumi, Katsuyuki Inoue, Minoru Yoshida,
MITSUBISHI ELECTRIC CORPORATION ADVANCED TECHNOROGY R&D CENTER

‡Kazuki Matsumura,
Graduate School of Informatics, Kyoto University

３．設計モデル生成処理

３．１　設計モデル生成処理概要

設計検証を実現するためには、設計資料として状態遷移モデルが用意されていることが望ましい。しかし、現実には設計者が状態遷移モデルの記法に習熟していなかったり、状態遷移モデルの作成に作業リソースを割くことができなかったりするため、状態遷移モデルが作成されないことも多い。設計検証を実開発に適用するためには、状態遷移モデルを簡単に作成する方法が重要となる。

そこで、原始モデルに対し設計知識を適用して設計モデルを生成する処理を支援するツールを提案する。ツールがユーザを支援することで、少ない作業量で設計モデルを生成することが可能となる。

３．２　設計モデル生成の例

ここで、ブレーキ監視装置用 S/W の設計モデル生成時の例を挙げる。

まず、ユーザは原始モデルとしてブレーキモデルを作成し、ツールに入力する。これは、ブレーキ監視装置用 S/W の設計モデルの種となるものである。図２に示すように、この例ではブレーキの故障および故障からの復帰が関心のある事象である。

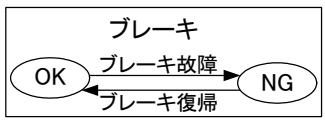

図２　ブレーキを表す原始モデル

次に、この原始モデルに適用する設計知識をユーザが選択する。まず、設計対象の構成を指定する。ここでは、ツールが提示する構成の中から、図３に示す監視装置と監視対象が一対一で繋がれた構成を指定する。

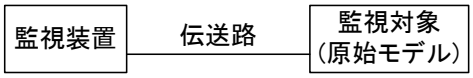

図３　設計対象の構成

設計対象の構成を指定することにより、選択可能な要求仕様がツールから提示される。ここでユーザは、”伝送路故障”と”データ整合性の確認”を要求仕様として選択する。ツール内部では、要求仕様と設計知識を関連付けて格納しているため、ツールは図４の設計知識を得ることができる。

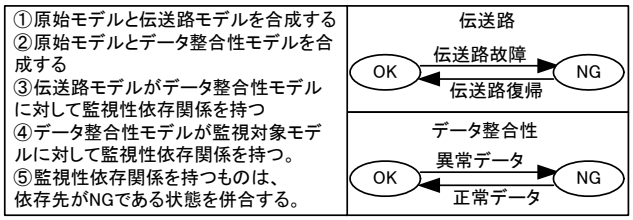

図４　設計知識

この設計知識には、原始モデルと合成するモデル(伝送路モデル・データ整合性モデル)および各モデル間の関係が記述されている。ここで、”モデル A がモデル B に監視依存関係を持つ”とは、モデル A が異常状態の場合、監視装置はモデル B の状態を監視できないことを意味する。

最後にツールが、設計知識を原始モデルに適用することで、図５に示す設計モデルを生成することができる。

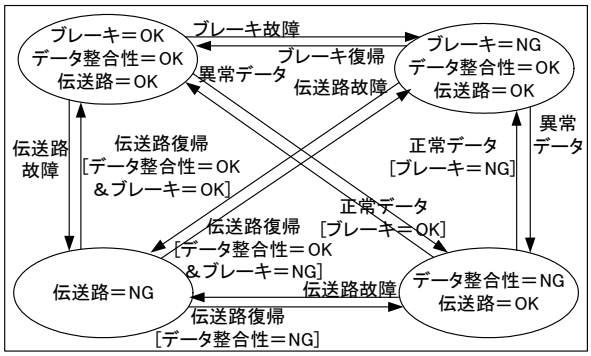

図５　ブレーキ監視装置用 S/W 設計モデル

この設計モデルは、伝送路およびデータ整合性が正常の場合のみブレーキ状態を監視することができることを表している。

このようにツールが設計知識の適用を支援することで、検証の対象となる設計モデルを比較的容易に作成することができるようになる。

４．検証モデル生成処理

４．１　検証モデル生成処理概要

現状では UML ステートマシン図を直接入力可能なモデル検査ツールは存在しない。設計検証の実現のためには、作成した設計モデルを検証モデルへ変換する必要がある。その際のユーザの作業量軽減のために、生成規則を用いて設計モデルから検証モデルを自動生成するツールを提案する。

４．２　検証モデル生成規則

生成規則は、各モデル検査ツールに対応したものであるため、モデル検査ツールごとに個別に用意する必要がある。一般的には、イベント・アクション・状態・遷移について UML ステートマシン図からモデル検査ツールの入力形式への変換方法を規則化することで定義される。

４．３　試作ツール

検証モデル生成処理を実行するツールの試作を行った。ツールは、内部にモデル検査ツール SMV 用の生成規則を保持しており、入力された UML ステートマシン図を SMV に入力可能な検証モデルに変換する機能を持つ。また、複数の設計モデルの並行動作についての検証を行うために、複数の UML ステートマシン図を一つの検証モデルに変換する機能も持つ。ツールは Eclipse[5] のプラグインを用いて実現した。ツールの画面イメージを図６に示す。

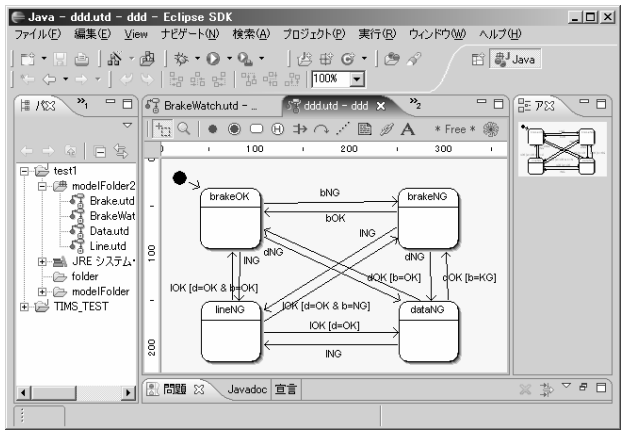

図６　試作ツール画面

ユーザは、UML ステートマシン図である設計モデルを入力する。入力には EclipseUML[6]を用いる。入力後、変換処理を実行すると SMV に対応した検証モデルを得ることができる。ツールから得られる検証モデルに、時相論理式を用いて記述された検証式を付加することで SMV を用いた検証が可能となる。

本ツールによって、UML ステートマシン図をモデル検査ツールに入力するための負荷が軽減された。

５．おわりに

S/W の設計検証にモデル検査ツールを実適用する方法について提案した。ここでは、ユーザに以下の二つの処理を実行するシステムを提供することにより、設計検証における設計モデルの生成からモデル検査の実行までの処理を簡単に実行する機能を提供する。

・設計モデル生成の支援
・設計モデルから検証モデルの自動生成

今後の課題として、設計モデル生成支援ツールの実装、検証モデルに対する検証式の入力支援方法の検討などが挙げられる。

参考文献

[1]K.L.McMillan.:Symbolic Model Checking, Kluwer- Academic, 1993.
[2]G.J.Holzmann.:The SPIN Model Checker Primer and Reference Manual, Addison-Wesley, 2004.
[3]中島震:モデル検査検証のソフトウェア開発への応用,日本ソフトウェア科学会 DSW2004, February 2004.
[4]OMG:UNIFIED MODELING LANGUAGE, http://www.uml.org
[5]Eclipse.org:Eclipse, http://www.eclipse.org/
[6]Omondo:EclipseUML, http://www.omondo.com/