7L−4　到達可能性解析に基づく通信ソフト検証方法の提案

西木 健哉*　宮崎 聡*　柳生 和男*　馬嶋 宏*　熱海 史郎**
*(株) 日立製作所　**(株) 日立ソフトウェアエンジニアリング

1. はじめに

通信系ソフトは状態遷移表を中心に設計されており、その正当性を確認する作業が重要である。本稿では、複数の状態遷移モデル間の仕様にデッドロック、無限ループなどの論理的誤りが発生しないかどうかを調べるプロトコル検証技法（到達可能性解析）[1][2]を拡張して、実装環境を考慮して作成された状態遷移表にも適用可能な検証方法を提案する。さらに、状態遷移仕様データを表形式のまま入力できることを特徴とする検証ツールプロトタイプの概要を述べる。

2. 仕様検証方法

2.1 解析モデル

（1）基本モデル

従来の拡張型有限状態機械EFSM(Extended Finite State Machine)モデルに従う。即ち、通信システムを構成するプロセスをEFSMとしてモデル化し、その動作を状態遷移表で表現する。プロセス同士は論理的な全二重チャネルを介して、通信の基本単位であるメッセージ（サービスプリミティブ、プロトコルデータ単位等）を授受することにより相互作用を行なう。

（2）基本モデルの拡張

上記モデルでは通信管理プログラムの実装環境を十分モデル化できないため、基本モデルを拡張して以下に挙げる処理を扱えるようにした（図1参照）。

(a) プロセスの動作レベル

動作レベルの高いプロセスから順に、状態遷移の処理を実行する。

(b) 2種類のメッセージ処理

・キューイングモード

プロセス切り換えに相当する。送出メッセージはチャネル上のFIFOキューにスケジュール後、順次処理する。

・ランデブモード

サブルーチンコールに相当する。送出メッセージは直ちに相手側プロセスで処理する。

(c) 優先データ転送

同一チャネルに複数の優先度付きキューを設定し優先度の高いキューから順に受信処理する。

(d) チャネル擬似障害

・コネクション型モード（回線障害）

ランダムなタイミングでキュー内の全てのメッセージが削除され、チャネルが障害から回復するまでメッセージは転送できない。

・コネクションレス型モード（伝送エラー）

ランダムなタイミングでキュー内の任意のメッセージが削除され、それ以外は正常に送受信される。

(e) タイマ制御

プロセス単位に設定し、起動／停止処理を受け付ける。起動されたタイマは任意のタイミングでタイムアウトの割り込みを発生する。

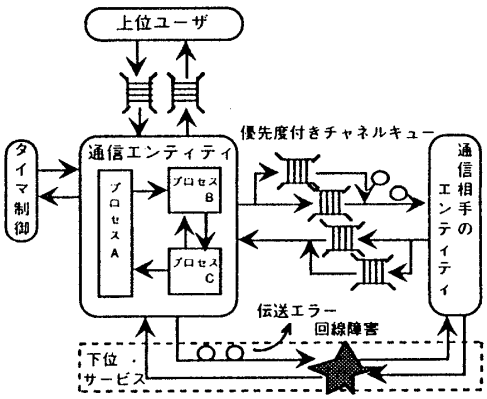

図1　解析モデルの模式図

2.2 検証機能

従来の検出項目である未定義入出力、デッドロック、無限ループ、チャネル容量オーバフロー、および実行不可遷移と併せて、以下の機能を追加した。

（1）異常終了処理実行の検出

状態遷移表における異常終了処理は、他プロセスとのインタフェース不正など内部矛盾の検出時に、起動中プログラムを停止することである。もし異常終了処理が到達可能性グラフに現われるならば、内部矛盾の有無にかかわらず異常終了してしまう可能性がある。そこで異常終了処理の実行を検出し、シーケンスの見落としや状態遷移条件の漏れを防止する。

（2）タイマ,フラグ設定誤りの検出

タイマ起動中にさらに起動処理が実行された場合、もしくは停止中にさらに停止処理が実行された場合を

A Verification Method for Communication Software Based on Reachability Analysis

Ken'ya NISHIKI*, Satoshi MIYAZAKI*, Kazuo YAGYU*, Hiroshi MAJIMA*, Shirou ATSUMI**
*HITACHI ,Ltd., **HITACHI Software Engineering Co., Ltd.

二重処理として警告する。また、初期状態に戻った時点でのリセット忘れに警告を出す。同様に、フラグの二重処理についても警告を出す。

（3）特定シーケンスの存在確認

与えられた初期状態から出発して、期待する状態に到達するシーケンスが存在するかどうかを確認する。例えば、初期状態（コネクションなし）からデータ転送状態（コネクション確立）に至るシーケンスの存在有無である。

3. 検証ツールの概要

提案方法に基づき検証ツールプロトタイプを作成した。本ツールの特徴は、図2に示すようにコンパイル可能なソースコードを生成するツールへ入力する状態遷移仕様をターゲットとして、論理的な整合性の検証を機械的に行う点である。これにより、プロトコルレベルの整合性ではなく、実装されるソフトウェアの整合性を保証することができる。さらに表形式データを直接入力して解析することにより、検証データを作成する負担を軽減できる。以下に、入力データの定義方法、検証結果の出力、およびプロトタイプの評価について述べる。

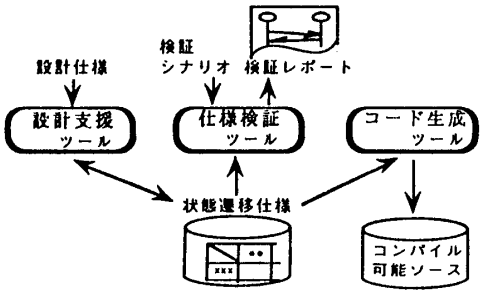

図2　ツール構成

3.1 入力定義方法

（1）状態遷移仕様

図3に示すような状態遷移表形式言語により以下の項目を記述する。

・状態遷移表（枠の定義、プレディケートの呼び出し、アクションの呼び出し、次ステータスの呼び出し）
・プレディケート文（データ形式のチェック、リトライ回数、バッファ確保などの条件判定式とプレディケートを得るために実行される前処理）
・アクション文（メッセージ送受信やタイマ設定／解除などの状態遷移において実行する処理）

（2）検証シナリオ

検証対象モジュールの構成情報として以下の項目を定義する。

・遷移処理優先度，　・タイマ／フラグ／変数
・内部事象（状態遷移表の記述のない上位層からの要求／応答を含む）
・解釈可能な形式で表現したアクションおよびプレディケート処理

```
*--------*----------*---------*---------*
| STATE  |  NEUTRAL |  OPEN   | CLOSING |
*--------+          |         |         |
| EVENT  |          |         |         |
*--------+----------+---------+---------+
| T-OPEN-REQ | ?GETCTL     | <ABEND> | <ABEND> |
|            | <GET-MIB-REQ> | -> *  | -> *    |
|            | ->OPEN      |         |         |
|            +-------------+         |         |
        .....
ACTION
 <GET-MIB-REQ> (* NEUTRAL ジョウタイデT-OPEN-REQウケツケ*)
                /* セイギョテーブルヲカクホスル */
        .....
 <ABEND>
        .....
ENDACTION
PREDICATE
 ?GETCTL:   (*セイギョテーブルヲカクホシタ*)
        .....
ENDPREDICATE
```

図3　状態遷移仕様の記述例

・出力メッセージと受信側で発生するイベントとの対応関係

検証実行時の制御情報として以下の項目を定義する。

・メッセージ処理モード，　・擬似障害発生条件
・チャネルキュー容量，　・データ転送優先度
・初期／終端グローバル状態

3.2 検証結果の出力

誤りの検出された箇所とその誤りを除去するために修正すべき箇所は一致しないことが多い。その場合、設計者は検証結果を参照しながら誤りの原因を特定しなければならない。本ツールでは、初期状態から誤りに至るまでのシーケンス図を出力することにより修正作業を支援する。

3.3 プロトタイプ評価

検証ツールプロトタイプをOSIトランスポートレイヤの設計仕様などに適用し、検証機能および性能を確認した。従来人手で行なっていた設計仕様の検証を本ツールで支援することにより、仕様設計レベルでの作り込み不良を早期に、効率よく摘出することができ、設計仕様の品質向上と後戻り工程の削減につながる。

4. おわりに

通信ソフトの設計誤りを仕様レベルで検出する一手段として、到達可能性解析の拡張による設計仕様検証方法を提案し、提案方法に基づく検証ツールプロトタイプを開発した。

参考文献

[1] Zafiropulo,P.,et al.: Towards analyzing and synthesizing protocols,IEEE Trans.Comm.,COM-28(4), 651-660 (1980)

[2] 白鳥則郎他：EXPA：パータベーション解析に基づく通信プロトコルの検証法,情処学論,Vol.26(3), 446-453 (1985)