情報処理学会論文誌 Vol.57 No.4 1262–1273 (Apr. 2016)

逸脱分析を用いた要求仕様書からのテスト項目抽出手法

大林 英晶[1,a)] 森崎 修司[1] 渥美 紀寿[1,b)] 山本 修一郎[1,c)]

受付日 2015年5月28日, 採録日 2016年1月12日

概要：網羅的かつシステマチックにテスト項目を得ることを目的とし，要求仕様書の原文を分析しテスト項目を得る手法を提案する．具体的には，要求仕様書の原文を IEEE 29148 の定義に沿って，主体，対象，機能，条件，制約といった要素に分解し，要素ごとに逸脱を分析する．得られた逸脱において期待する動作をするかどうかを確かめるテスト項目を得る．提案手法により実際に要求仕様書からテスト項目が得られるか博士前期課程の学生を被験者として実験した．教育目的で作成され広く活用されている架空の電気ポットの仕様書 “話題沸騰ポット” 第 7 版が対象である．実験の結果，58 項目のテスト項目が得られた．また，これらのテスト項目の妥当性を確認するため，同仕様書から提案手法によらず実務者が作成したテスト項目と比較したところ，機能テストにおいて 50 項目のテスト項目が共通していた．

キーワード：テスト項目，要求記述表，HAZOP

An Approach for Extracting Test Items Using Requirements Specification Description

Hideaki Ohbayashi[1,a)] Shuji Morisaki[1] Noritoshi Atsumi[1,b)] Shuichiro Yamamoto[1,c)]

Received: May 28, 2015, Accepted: January 12, 2016

Abstract: This paper proposes an approach for extracting test items from requirement specification document. The approach divides each statement in requirement specification document into elements defined by IEEE 29148 such as subject, object, function, condition, constraint. The approach analyzes hazardous conditions by combining the obtained elements and guide words. The approach extracts test items with the hazardous conditions. We conducted an experiment with a graduate student to empirically evaluate the approach. The requirement specification document in the experiment is a publicly available document for engineers' education. The result of the experiment indicates that the student extract 58 test items from the requirement specification and that the extracted test items had 50 common items with the test items extracted by practitioner team from the same specification document.

Keywords: test items, requirement decomposition, HAZOP

1. はじめに

近年ソフトウェアシステムは我々の社会基盤を構築する重要なシステムとなっており，システムの障害は社会基盤を揺るがす深刻な事態に陥る可能性がある．システム障害の発生リスクを明らかにしたり，ソフトウェアが期待どおり開発されていることを評価する評価項目を検討したりする必要がある．しかし，ソフトウェアの規模は増大する一方であり，網羅的に発生リスクや評価項目を明らかにするのは困難になりつつある．

障害の発生リスクを網羅的に明らかにする手法として FTA (Fault Tree Analysis) [12], FMEA (Failure Mode and Effects Analysis) [13], HAZOP (Hazard and Operability Analysis) [14] といった手法がある．これらの手法は，元々はソフトウェアを対象としたものではないため，ソフトウェアにおいても適用できるよう Software FMEA [16] や HAZOP のガイドワードをソフトウェアに適用しやすいように整理した SHARD [4]，要求仕様書に HAZOP のガイドワードを適用し要求からの逸脱のリスクを検討する手法 [11]，ソフトウェアプロダクトラインプロセスの安全性

[1] 名古屋大学
Nagoya University, Nagoya, Aichi 464–8601, Japan
[a)] oobayashi.hideaki@d.mbox.nagoya-u.ac.jp
[b)] atsumi@nagoya-u.jp
[c)] yamamotosui@icts.nagoya-u.ac.jp

を HAZOP で分析する手法 [6]，保証ケースに基づくテスト十分性の合意形成手法 [24] が提案されている．しかし，これらの手法は障害の発生リスクを明らかにしているものの，ソフトウェアが期待どおり開発されているか評価する評価項目を得る方法には言及がない．

ソフトウェアを対象とし，要求仕様書からテスト項目を得る研究として，Sarma らの研究 [19]，Reza らの研究 [17] がある．Sarma らは要求仕様書から作成したユースケース図，シーケンス図を組み合わせ，System Testing Graph（STG）と呼ぶ図を作成し，STG からテスト項目を作成する手法を提案している．STG はユースケース図から作成したイベントフローを基に対応するシーケンス図から読み取り作成する．得られるテスト項目はイベントフローどおりにプログラムが実行されるかどうかを確かめるものである．Reza らは，Web アプリケーションの要求仕様書から状態遷移図を得て，状態遷移図からテスト項目を生成する手法を提案している．状態遷移図を用いることによりすべての遷移，経路，状態を網羅するテスト項目が作成できる．Sarma らの手法と同様に得られるテスト項目はホワイトボックステストであり，シーケンス図や状態遷移図に誤りがあると正しいテスト項目が得られない．本研究では，要求仕様書から逸脱を得て，ブラックボックステストのテスト項目を得るため，設計が誤っている場合でも正しいテスト項目が得られる．

与えられたパラメータと値の組合せを自動生成し，なるべく少数のテスト項目で組合せを網羅することを目的とした研究として，Jacek らの PICT [10] や Cohen の研究 [3] がある．PICT [10] は与えられたパラメータと値の組合せを自動生成し，必要のない組合せを省いた低位レベルテストケースを得る．Cohen は，与えられたパラメータと値の組合せから，*k-way* テストの最小のテスト集合を作成する最適化問題として定義し，探索的手法を組み合わせて低位レベルテストケースを作成する手法を提案している．*k-way* テストは任意の k 個のパラメータの組合せを網羅するテストケースである．本研究は，これらの手法の入力となるテスト項目（高位レベルテストケース）を得るため，組み合わせて使うことができる．

本論文では逸脱状態でもソフトウェアが期待どおり動作するか確かめるための評価項目を得るための手法を提案する．具体的には，要求仕様書の原文を分析し，制約や条件を取り出し，ガイドワードを適用し，逸脱を網羅的に得る．得られた逸脱からテスト項目を得る．提案手法により要求仕様書から実際にテスト項目が得られるか実験により確かめる．また，テスト項目の網羅性を評価するため，同一の要求仕様書から提案手法によらず得られたテスト項目と比較する．

以降，2 章で提案手法を述べ，3 章で要求仕様書を用いた適用例を説明する．4 章で実験と結果を述べ，5 章で考察し，6 章でまとめる．

2. 提案手法

2.1 手順

提案手法は要求仕様書から要素を取り出し，個々の要素の逸脱を網羅的に列挙する．要素は文献 [23] で定義されている，自然言語の文中の主語，動詞，補語である．得られた逸脱においてエラー処理をはじめとする対処があるかどうかを確認するためのテスト項目を得る．逸脱は要求仕様書が前提としている状態から外れる状態である．逸脱を網羅的に列挙するために使用するキーワードをガイドワードと呼ぶ．

提案手法は次の手順から構成される．

手順 1：要求仕様書の原文を IEEE 29148 [23] で定義されている要素に分解し，要求記述表に記述する．

手順 2：要求記述表の各要素にガイドワードを組み合わせて逸脱を得る．

手順 3：逸脱に対応するテスト項目を作成する．

次節以降で各手順の詳細を述べる．

2.2 要求記述表の作成（手順 1）

要求仕様書の文に含まれる主体 O，対象 S，条件 C，機能 F，制約 N を取り出し，**表 1** の形式の要求記述表を作成する．具体的には，要求仕様書の文 $\mathcal{R}$ から，IEEE 29148 で定

表 1 要求記述表の構成
Table 1 Elements of the requirement description table.

要求仕様：R_i				
主体 S_i	対象 O_i	条件 C_i	機能	制約 N_i
前提条件 Sp_i	前提条件 Op_i	主体状態 Cs_i	機能内容 F_i	応答制約 Nr_i
		対象状態 Co_i		出力制約 No_i
		イベント条件 Ce_i		値制約 Nv_i
		イベント入力 Ct_i		

義されている4つの構文型（$CSONF$型，CFN型，CNN型，SCF型）にあてはまる文$R=(R_1,\cdots,R_i,\cdots,R_n)$を選ぶ．$\mathcal{R}$のうち，4つの構文型にあてはまらない文$R'$は要求記述表の作成には使わない．

4つの構文型は以下のとおりである．

- $CSONF$型
  ［条件C］で［主体S］が［対象O］を［制約N］で［機能F］する必要がある．
- CFN型
  ［条件C］で［機能F］は［制約N］である必要がある．
- CNN型
  ［条件C］で［制約N_1］は［制約N_2］である必要がある．
- SCF型
  ［主体S］が［条件C］で［機能F］する必要がある．

要求仕様書の文R_iの要素S，O，F，C，Nから下の手順で要求記述表T_iを得る．

要求記述表の個々の項目は下の(1)〜(5)の手順で得る．なお，本説明の例として次の$CSONF$型の要求文を使う．

> システムにアップデートがあり[Cs]、バッテリーが50%以上残っている状態[Co]でパスワードを入力[Ct]しユーザ認証を行ったとき[Ce]、アップデートマネージャ[S]はシステム[O]に対して5秒以内[Nr]にアップデート画面[No]を1回[Nv]表示しアップデートを開始する[F]。

(1) 主体名称S_iと前提条件Sp_i

システムの構成要素である主体Sの記述から主体の名称S_iとSの実行の前提条件Sp_iを得る．Sの実行の前提条件Sp_iは他の要求文R_j（$i\neq j$）から得たり，要求文として明示されていない前提条件を補足したりして記述する．例文では，S_iは“アップデートマネージャ”である．R_jに“システム起動10秒後からアップデートマネージャの動作を開始する.”といった記述がある場合，“アップデートマネージャはシステム起動後10秒後に動作を開始する”をSp_iに記述する．R_iに複数の構成要素がまとめて記述してある場合はそれらすべてをS_iとする．Sの実行に前提条件がないと判断された場合には$Sp_i=\emptyset$とし，要求記述表のSp_iは空欄とする．

(2) 対象O_iと前提条件Op_i

Sの操作対象Oの記述からOの名称O_iとOの資源（データ，被制御装置等）が利用できる前提条件Op_iを得る．O_iがない場合にはOp_iは$\emptyset$である．O_iはあるが，前提条件がない場合にはO_iをOp_iとする．なお，前提条件Op_iは他の要求文R_j（$i\neq j$）から得る場合と要求文として明示されていない前提条件を補足して記述する場合がある．例文では，O_iは“システム”である．R_iにはO_iの前提条件がないため，Op_iはO_iと同じ“システム”となる．もし，R_jに“アップデート中はシステムはスリープ状態にならない”があった場合，R_jから補足してOp_iは“アップデート中はシステムはスリープ状態にならない”となる．

(3) 機能内容F_i

R_iに記述してある主体S_iが対象O_iに対して行う振舞いを機能内容F_iとして文章で記述する．

(4) 条件C_i

条件Cを主体状態Cs_i，対象状態Co_i，イベント入力Ce_i，入力条件Ct_iに分けて文章で記述する．$C_i=(Cs_i,Co_i,Ce_i,Ct_i)$とし，$C_i$の各要素に該当する記述が$R_i$にない場合には$\emptyset$とする．$Cs_i$は主体$S_i$の状態に関する条件を文章で記述する．$Co_i$は対象$O_i$の状態に関する条件を文章で記述する．$Ce_i$と$Ct_i$はイベントに関する条件である．$Ce_i$は機能の実行契機であり，その際に与えられる入力があればCt_iとして記述する．

例文Rでは，“システムにアップデートがあり”は主体S（アップデートマネージャ）の状態であるためCsとなる．“バッテリーが50%以上残っている状態”は対象O（システム）の状態であるためCoである．“ユーザ認証を行ったとき”は機能F（アップデートを開始する）の実行契機であるためCeである．“パスワードを入力”はCeに与えられる入力であるためCtである．

(5) 制約N_i

制約N（CNN型の場合，N_1，N_2）を応答制約Nr_i，値制約Nv_i，出力制約No_iに分けて文章で記述する．$N_i=(Nr_i,Nv_i,No_i)$とし，N_iの各要素に該当する記述がR_iにない場合には$\emptyset$とし，要求記述表は空欄とする．Nr_iは機能の応答時間に関する制約である．Nv_iは機能の入出力と内部で保持する値に関する制約である．No_iは出力制約であり，機能の出力先や出力サイズに関する制約である．CNN型の場合，N_2は値制約Nr_iであり，N_1はそれ以外の制約である．

例文Rの“5秒以内にアップデート画面を1回表示しアップデートを開始する.”において，“5秒以内”は機能F（アップデートを開始する）の応答時間に関する制約であるためNrである．“アップデート画面”は機能Fの出力先の制約であるためNoである．“1回表示”は機能Fの入出力に関する制約であるためNvである．

4つの構文型から得られるT_iはそれぞれ以下のようになる．

- $CSONF$型
  $T_i=(S_i,Sp_i,O_i,Op_i,F_i,C_i,N_i)$

表 2 要素とガイドワードのグループ
Table 2 The groups of the elements and the guidewords.

グループ	要求記述表 T_i の要素	ガイドワード G
値	Nv_i，No_i	g_1 なし，g_2 以外，g_3 部分（一部）
タイミング	Nr_i	g_1 なし，g_4 冗長，g_5 遅い，g_6 早い
動作	Sp_i，Op_i，Cs_i，Co_i，Ce_i，Ct_i	g_1 なし，g_2 以外，g_3 部分（一部），g_4 余分（冗長）
機能	F_i	g_1 なし

- CFN 型
  $T_i = (F_i, C_i, N_i)$
- CNN 型
  $T_i = (C_i, N_i)$
- SCF 型
  $T_i = (S_i, Sp_i, F_i, C_i)$

2 つの要求文 R_i，R_j が互いに影響を与えるような場合には，1 文にまとめるか Sp_i，Op_i，Sp_j，Op_j に追加する．たとえば，R_i：システムが信号 X を受信したときに警告音を鳴らす，R_j：システムが信号 Y を受信したときには警告音を鳴らす，という要求文があるときには，R_k：信号 X，Y のいずれかまたは両方を受信したときには警告音を鳴らす，として 1 文にまとめる．まとめられない場合には，関係する要求記述表の Sp，Op に記入する．たとえば，R_i と R_j を 1 文にまとめず，T_i，T_j を作成する場合，T_i の Sp には，システムが信号 Y を受信しない，T_j の Sp には，システムが信号 X を受信しない，と記述する．

2.3 ガイドワードによる逸脱分析（手順 2）

T_i の各要素にガイドワード G を組み合わせて逸脱 D_i を得る．組み合わせるガイドワードは要素によって異なり，**表 2** に示すグループから決まる．要素とガイドワードの組合せは 4 つのグループ（“値”，“タイミング”，“動作”，“機能”）に分かれている．値，タイミング，動作，機能のそれぞれの観点において，各要素が要求文の前提から外れるものをガイドワードを使いながら網羅的に列挙し，逸脱とする．たとえば，要素 Nv_i（値制約）の場合，Nv_i のグループ“値”のガイドワード g_1，g_2，g_3 と組み合わせて，要求文の前提から外れるような値や値の条件を逸脱として列挙する．要素とガイドワードの対によって得た逸脱を Nv_ig_1，Nv_ig_2，Nv_ig_3 とする．なお，要素とガイドワードの組合せは，この 4 つの観点で分けている．値，動作，機能の観点は，Leveson の文献 [14] で提案されたガイドワードから観点に適切なもの（None，As well as，Part of，Other than）を選んだ．タイミングの観点は，McDermid の文献 [15] で提案されたガイドワード Late，Early を選んだ．

得られる逸脱 D_i は R_i の型によって異なり，以下のとおりである．

- $CSONF$ 型
  $D_i = (Sp_iG, Op_iG, F_iG, C_iG, N_iG)$
- CFN 型
  $D_i = (F_iG, C_iG, N_iG)$
- CNN 型
  $D_i = (C_iG, N_iG)$
- SCF 型
  $D_i = (Sp_iG, F_iG, C_iG)$

ただし，R_i から逸脱として起こりえないと判断できるものは除外する．

グループごとの逸脱は以下のように得る．“値” グループの要素である値制約 Nv_i，出力制約 No_i はガイドワード g_1，g_2，g_3 と組み合わせる．たとえば，Nv_i が “エアコンの設定温度は 20 から 30 度” であるとすると，Nv_ig_1 は “エアコンの設定可能温度は 20 から 30 度ではない”，Nv_ig_2 は “エアコンの設定可能温度は 20 から 30 度以外”，Nv_ig_3 は “エアコンの設定可能温度は 20 から 30 度の範囲の一部” である．

“タイミング” グループの要素である応答制約 Nr_i はガイドワード g_1，g_4，g_5，g_6 と組み合わせる．たとえば，Nr_i が “1 秒以内に（確認音を鳴らす）” であるとすると，Nr_ig_1 は “1 秒以内でない”，Nr_ig_4 は “1 秒以内と余分に”，Nr_ig_5 は “1 秒以内より遅い”，Nr_ig_6 は “1 秒以内より早い” である．ただし，ガイドワードによって “1 秒以内より早い” という逸脱が得られるが，1 秒以内であればよいので，早いことは逸脱に該当しない．

“動作” グループの要素である主体前提条件 Sp_i，対象前提条件 Op_i，主体条件 Cs_i，対象条件 Co_i，イベント条件 Ce_i，入力条件 Ct_i はガイドワード g_1，g_2，g_3，g_4 と組み合わせる．たとえば，Ce_i が “（リモコンから）電源 ON/OFF 信号を受信したとき” であるとすると，Ce_ig_1 は “電源 ON/OFF 信号を受信しないとき”，Ce_ig_2 は “電源 ON/OFF 信号以外を受信したとき”，Ce_ig_3 は “電源 ON/OFF 信号を一部受信したとき”，Ce_ig_4 は “電源 ON/OFF 信号と余分な信号を受信したとき” である．

“機能” グループの要素 F_i はガイドワード g_1 と組み合わせる．たとえば F_i が “確認音を鳴らす” であるとすると，F_ig_1 は “確認音が鳴らない” である．

2.4 テスト項目の作成（手順 3）

テスト項目は T_i から作成する正常系のテスト項目 I_{T_i} と D_i から作成する異常系のテスト項目 I_{D_i} である．F_i のな

い CNN 型の R_i から作成した T_i は I_{T_i} を作成する際には除外する．I_{T_i}，I_{D_i} の記述内容は文献 [8] の高位レベルテストケース（high-level test case）に該当する．

I_{T_i} は T_i から得る．Sp_i，Op_i，C_i，N_i の条件，または，制約のもとで S_i，O_i，F_i が期待どおりかどうかを確認するテスト項目を I_{T_i} とする．たとえば，R_i が “エアコンは電源 ON/OFF 信号を受信したとき確認音を鳴らす” とすると，F_i は “確認音を鳴らす”，Ce_i は “電源 ON/OFF 信号を受信したとき” である．作成するテスト項目 I_{T_i} は “電源 ON/OFF 信号を受信したとき，確認音を鳴らすことを確認する” である．

テスト項目 I_{D_i} は逸脱 D_i から得る．I_{D_i} は F_iG を除く D_i' のもとで F_i が期待どおりか確かめるためのテスト項目である．逸脱 Sp_iG，Op_iG，C_iG，N_iG に対応するテスト項目を作成する．逸脱を組み合わせたテスト項目の作成はしない．先述の R_i の場合，作成するテスト項目 I_{D_i} は “電源 ON/OFF 信号を受信しないとき確認音が鳴らないことを確認する（Ce_ig_1）”，“電源 ON/OFF 信号以外を受信したとき確認音が鳴らないことを確認する（Ce_ig_2）”，“電源 ON/OFF 信号を一部受信したとき確認音が鳴らないことを確認する（Ce_ig_3）”，“電源 ON/OFF 信号と余分な信号を受信したとき確認音が鳴らないことを確認する（Ce_ig_4）” となる．

得られた I_{T_i}，I_{D_i} をもとにテストの実行に必要なパラメータや予測結果を追加し，低位レベルテストケース（low-level test case）としたうえでテストを実行する．高位レベルテストケースから低位レベルテストケースは文献 [9] にあるような Pairwise testing [18] をはじめとする組合せテスト技法により得る．なお，2 つの異なる逸脱 D_i，D_j（$i < j$）から一部重複するテスト項目 I_{D_i}，I_{D_j} を得ることがある．その場合には I_{D_j} の重複するテスト項目を削除する．

3. 適用例

以下の船のレーダシステムの要求仕様書の一部（$\mathcal{R}$）を用いて説明する．

> システムの要求仕様を以下に述べる．システムは電源の供給があれば，稼働し続ける．信号 x を受信したとき，システムは信号 x の受信ビットを 2 秒以内で設定する必要がある．

3.1 要求記述表の作成（手順 1）

$\mathcal{R}$ を構文型に分類すると以下のようになる．

- 1 文目：システムの要求仕様を以下に述べる．
- 2 文目：システムは［主体 S］電源の供給があれば［条件 C］，稼働し続ける［機能 F］．
- 3 文目：信号 x を受信したとき［条件 C］，システムは［主体 S］信号 x の受信ビットを［対象 O］2 秒以内で［制約 N］設定する［機能 F］必要がある．

1 文目は提案手法では対象外である要求 R' である．2 文目は SCF 型の文 R である．3 文目は $CSONF$ 型の文 R である．以降では 2 文目を R_1，3 文目を R_2 とする．

S_1 は “システム”，C_1 は “電源の供給があれば”，F_1 は “稼働し続ける” である．C_1 は主体の状態であるため，“電源の供給があれば” は Cs_1 である．

S_2 は “システム”，O_2 は “信号 x の受信ビット”，C_2 は “信号 x を受信したとき”，F_2 は “設定する”，N_2 は “2 秒以内” である．S_2 の前提条件は T_1 の Cs_1 であるため “電源の供給があれば” は Sp_2 である．O_2 は対象の前提条件であるため “信号 x の受信ビット” は Op_2 である．C_2 はイベントの条件であるため “信号 x を受信したとき” は Ce_2 である．N_2 は応答時間に関する制約であるため “2 秒以内” は Nr_2 である．

R_1 から得られる T_1 は $T_1 =$ (システム (S_1), $\emptyset(Sp_1)$, (電源の供給があれば $(Cs_1), \emptyset(Co_1), \emptyset(Ce_1), \emptyset(Ct_1)$)，稼働し続ける (F_1)) である．対応する要求記述表を**表 3** に示す．

R_2 から得られる T_2 は $T_2 =$ (システム (S_2), 電源の供給があれば (Sp_2), 信号 x の受信ビット (O_2), 信号 x の受信ビット (Op_2), 設定する (F_2), $(\emptyset(Cs_2), \emptyset(Co_2)$, 信号 x を受信したとき $(Ce_2), \emptyset(Ct_2))$，(2 秒以内 $(Nr_2), \emptyset(Nv_2), \emptyset(No_2))$ である．対応する要求記述表を**表 4** に示す．

3.2 ガイドワードによる逸脱分析（手順 2）

T_1 から逸脱 D_1 を得る．得られる逸脱 D_1 は $D_1 = (Sp_1G, C_1G, F_1G)$ である．T_1 では Sp_1 は $\emptyset$，C_1 は $(Cs_1, \emptyset, \emptyset, \emptyset)$ であるため，$D_1 = (\emptyset, (Cs_1G, \emptyset, \emptyset, \emptyset), F_1G)$ となる．さらに，$\emptyset$ を消去し，G を対応する $g_1, \cdots, g_6$ に置き換えると，$D_1 = (Cs_1g_1, Cs_1g_2, Cs_1g_3, Cs_1g_4, F_1g_1)$ となる．D_1 の詳細を**表 5** に示す．

T_2 から逸脱 D_2 を得る．得られる逸脱 D_2 は $D_2 = (Sp_2G,\ Op_2G,\ F_2G,\ C_2G,\ N_2G)$ である．T_2 では C_2 は $(\emptyset, \emptyset, Ce_2, \emptyset)$，$N_2$ は $(Nr_2, \emptyset, \emptyset)$ であるため，$D_2 = (Sp_2G,\ Op_2G,\ F_2G,\ (\emptyset, \emptyset, Ce_2G, \emptyset),\ (Nr_2G, \emptyset, \emptyset))$ となる．さらに，$\emptyset$ を消去し，G を対応する $g_1, \cdots, g_6$ に置き換えると，$D_2 = (F_2g_1,\ Sp_2g_1,\ Sp_2g_2,\ Sp_2g_3,\ Sp_2g_4,\ Op_2g_1,\ Op_2g_2,\ Op_2g_3,\ Op_2g_4,\ Ce_2g_1,\ Ce_2g_2,\ Ce_2g_3,\ Ce_2g_4,\ Nr_2g_1,\ Nr_2g_4,\ Nr_2g_5,\ Nr_2g_6)$ となる．D_2 の詳細を**表 6** に示す．Nr_2g_6 の “2 秒以内より早い” はガイドワードによって得られた逸脱であるが，2 秒以内であればよいので，Nr_2g_6 は逸脱に該当しない．

3.3 テスト項目の作成（手順 3）

テスト項目 I_{T_1} を求める．S_1 は “システム”，F_1 は “稼働し続ける”，Cs_1 は “電源の供給があれば” である．テス

表 3 適用例の要求記述表 T_1
Table 3 Requirement description table T_1.

要求仕様：システム［主体（S)］は電源の供給があれば［主体状態（Cs)］稼働し続ける［機能（F)］				
主体 S_1: システム	対象 O_1:	条件 C_1	機能	制約 N_1
前提条件 Sp_1	前提条件 Op_1	主体状態 Cs_1 電源の供給があれば 対象状態 Co_1 イベント条件 Ce_1 イベント入力 Ct_1	機能詳細 F_1: 稼働し続ける	応答制約 Nr_1 出力制約 No_1 値制約 Nv_1

表 4 適用例の要求記述表 T_2
Table 4 Requirement description table T_2.

要求仕様：信号 x を受信したとき［条件（C)］，システムが［主体（S）] 信号 x の 受信ビットを［対象（O)］2 秒以内で［制約（N)］設定［機能（F)］する必要がある．				
主体 S_2: システム	対象 O_2: 信号 x の受信ビット	条件 C_2	機能	制約 N_2
前提条件 Sp_2 電源の供給があれば （R_1 から引用）	前提条件 Op_2 信号 x の受信ビット	主体状態 Cs_2 対象状態 Co_2 イベント条件 Ce_2 信号 x を受信したとき イベント入力 Ct_2	機能詳細 F_2: 設定する	応答制約 Nr_2 2 秒以内 出力制約 No_2 値制約 Nv_2

表 5 ガイドワードを組み合わせて得た逸脱 D_1
Table 5 Deviations D_1 obtained by combining the guidewords.

	F_1 稼働し続ける	Cs_1 電源の供給があれば
g_1 なし	稼働し続けない	電源の供給がない
g_2 部分	-	一部電源の供給があれば
g_3 以外	-	電源の供給以外があれば
g_4 冗長	-	電源が余分に供給されれば
g_5 遅い	-	-
g_6 早い	-	-

表 6 ガイドワードを組み合わせて得た逸脱 D_2
Table 6 Deviations D_2 obtained by combining the guidewords.

	F_2 設定する	Sp_2 電源の供給があれば	Op_2 信号 x の受信ビット	Ce_2 信号 x を受信したとき	$Nr_2$2 秒以内
g_1 なし	設定されない	電源の供給がない	信号 x の受信ビットがなし	信号 x を受信しないとき	2 秒以内でない
g_2 部分	-	一部電源の供給があれば	信号 x の受信ビットの一部	信号 x を一部受信したとき	-
g_3 以外	-	電源の供給以外があれば	信号 x の受信ビット以外	信号 x 以外を受信したとき	-
g_4 冗長	-	電源が余分に供給されれば	信号 x の受信ビットと余分なビット	信号 x と余分な信号を受信したとき	2 秒以内と余分に
g_5 遅い	-		-	-	2 秒以内より遅い
g_6 早い	-		-	-	2 秒以内より早い

ト項目 I_{T_1} は “システムは電源の供給があれば稼働し続けることを確認する”，である．

テスト項目 I_{D_1} を求める．逸脱 D_1 と対応するテスト項目 I_{D_1} を**表 7** に示す．

テスト項目 I_{T_2} を求める．Sp_2 は “電源の供給があれば”，Op_2 は “信号 x の受信ビット”，F_2 は “設定する”，Ce_2 は “信号 x を受信したとき”，Nr_2 は “2 秒以内” である．テスト項目 I_{T_2} は “電源の供給があれば設定することを確認す

表 7 逸脱 D_1 と対応するテスト項目 I_{D_1}
Table 7 Test items I_{D_1} corresponding to deviations D_1.

逸脱 D_1	テスト項目 I_{D_1}
F_1g_1：稼働し続けない	-
Os_1g_1：電源の供給がない	電源の供給がないとき稼働し続けないことを確認する
Os_1g_2：一部電源の供給があれば	一部電源の供給があれば稼働し続けないことを確認する
Os_1g_3：電源の供給以外があれば	電源の供給以外があれば稼働し続けないことを確認する
Os_1g_4：電源が余分に供給されれば	電源が余分に供給されれば稼働し続けないことを確認する

表 8 逸脱 D_2 と対応するテスト項目 I_{D_2}
Table 8 Test items I_{D_2} corresponding to deviations D_2.

逸脱 D_2	テスト項目 I_{D_2}
F_2g_1：設定されない	-
Sp_2g_1：電源の供給がない	電源の供給がないとき設定されないことを確認する
Sp_2g_2：一部電源の供給があれば	一部電源の供給があれば設定されないことを確認する
Sp_2g_3：電源の供給以外があれば	電源の供給以外があれば設定されないことを確認する
Sp_2g_4：電源が余分に供給されれば	電源が余分に供給されれば設定されないことを確認する
Op_2g_1：信号 x の受信ビットがなし	信号 x の受信ビットがなしで設定されないことを確認する
Op_2g_2：信号 x の受信ビットの一部	信号 x の受信ビットの一部が設定されないことを確認する
Op_2g_3：信号 x の受信ビット以外	信号 x の受信ビット以外が設定されないことを確認する
Op_2g_4：信号 x の受信ビットと余分なビット	信号 x の受信ビットと余分なビットが設定されないことを確認する
Ce_2g_1：信号 x を受信しないとき	信号 x を受信しないとき設定されないことを確認する
Ce_2g_2：信号 x 以外を受信したとき	信号 x 以外を受信したとき設定されないことを確認する
Ce_2g_3：信号 x を部分受信したとき	信号 x を部分受信したとき設定されないことを確認する
Ce_2g_4：信号 x と余分な信号を受信したとき	信号 x と余分な信号を受信したとき設定されないことを確認する
Nr_2g_1：2 秒以内でない	2 秒以内でなく設定されないことを確認する
Nr_2g_4：2 秒以内と余分に	2 秒以内と余分に設定されないことを確認する
Nr_2g_5：2 秒以内より遅い	2 秒以内より遅く設定されないことを確認する

る”，“システムが信号 x の受信ビットを設定することを確認する”，“信号 x を受信したときシステムが信号 x の受信ビットを設定することを確認する”，“システムが信号 x の受信ビットを 2 秒以内に設定することを確認する” である．

テスト項目 I_{D_2} を求める．逸脱 D_2 と対応するテスト項目 I_{D_2} を**表 8** に示す．たとえば，Op_2g_1 の “信号 x の受信ビットがなし” に対応するテスト項目 I_{D_2} は “信号 x の受信ビットがなしで設定されないことを確認する” である．ただし，“2 秒以内に応答がなく設定されないことを確認する”，“2 秒以内より遅く設定されないことを確認する” は重複するため，前者のテスト項目は削除する．

4. 評価

4.1 概要

実験の対象となる要求仕様書 $\mathcal{R}$ は，話題沸騰ポット要求仕様書第 7 版 [20] とした．組込みソフトウェア管理者・技術者育成研究会（SESSAME）が作成したものである．架空の電気ポットの仕様書であるが，教育等の目的で広く使われており，複数回の改版を実施している．今回の評価の対象とした第 7 版はそれまでの版をもとに曖昧さをできる限り減らしたものであるとされている．本要求仕様書の一部を**表 9** に示す．対象とした要求仕様書は USDM（Universal Specification Describing Manner）[22] で記述されている．$\mathcal{R}$ は 48 文あり，そのうち R は 23 文である．表 9 では，“pot-270”，“pot-270-11”，“pot-270-21” といった ID が割り振られている文（ID の列のとなりの列の文）が R に該当する．

また，テスト項目の網羅性を確認するため，同要求仕様書を対象として実施されたテスト設計コンテスト 2013 [2] の優勝チームのテスト項目 [21]（I_{tc}）と比較する．I_{tc} の一部を**表 10** に示す．テスト項目は，表 10 中の “TSF01_11”，“TSF01_12”，“TSF01_13” といった ID が割り振られている文（ID の列のとなりの列の文）に該当する．テスト設計コンテストではソフトウェアテスト技術振興協会が開催し，実務者が同一の仕様書をもとにテスト項目を競う．テスト設計コンテストの成果物や評価基準は，本論文の評価の基準と同一ではないが，“テストすべきことがすべて盛り込まれているか” という基準があるため，網羅性を評価するための参考になると考えた．

提案手法を用いたテスト項目の作成は名古屋大学情報科学研究科の学生 1 名が行う．被験者は実務レベルのテスト項目作成の経験はないが，ソフトウェア開発やテストに関する知識がある．被験者には話題沸騰ポット要求仕様書第 7 版を渡し，テスト項目を作成してもらう．なお，要求

表 9 話題沸騰ポット要求仕様書第 7 版の一部（文献 [20] より引用）
Table 9 A Part of the target requirement specification (Cited from the article Ref. [20]).

要求	pot-270	タイマボタンを押すことで、時間を分でセットし、タイマを起動できる。
	理由	簡単な操作でタイマを操作したいから
	説明	タイマの用途として、カップラーメンを作る際の時間計測を想定している。
	pot270-11	コンセントに繋いだ直後は、0 min0 sec にリセットされ、タイマは停止した状態になる。
	pot270-21	タイマが起動している/していないにかかわらず、タイマボタンを 100 msec 以上押される度にタイムアップまでの残り時間の分に 1 分を加算し、秒の単位を 0 sec にクリアした値にセットし、セットした値（タイムアップまでの時間）を分単位のみで操作パネルのタイマ残り時間表示窓に表示する。【説明】59 min48 sec でタイマボタンを 1 回（100 msec）押したら、60 min0 sec をセットしたことになり、タイマ残り時間表示窓は 60 となる
	pot270-22	0 min0 sec から最大 60 min0 sec までセットすることができる。
	pot270-23	60 min0 sec のときに、更にタイマボタンを 1 回（100 msec）押されると、1 min0 sec をセットしたことになる。【説明】操作パネルには、1 → 2 → 3・・・・・・・→ 58 → 59 → 60 → 1 → 2 と表示される。

表 10 テスト設計コンテスト文書のテスト項目の一部（文献 [21] より引用）
Table 10 A Part of the test items in the test document Ref. [21].

タイマ機能	TRF01	タイマ機能の機能動作を検証する
	理由	タイマ機能が仕様通りに実装できているかどうかを検証するため
	説明	
	＜タイマの動作確認＞	
	TSF01_11	タイマ起動中の残り時間加算を確認する
		【説明】タイマボタン 100 msec 以上押下で表示が 1 分加算され、ブザーを 50 msec 鳴らす
	TSF01_12	タイマ停止中の残り時間加算を確認する
		【説明】タイマボタン 100 msec 以上押下で表示が 1 分加算され、ブザーを 50 msec 鳴らす
	TSF01_13	タイマボタンを押すのを止めた 1 sec 後からタイマ起動するのを確認する
		【説明】タイマが起動したことは、表示結果の変化で確認する（インシデント表）に関連
	TSF01_14	タイムアップの表示（0 分）・ブザー（100 msec 間隔で 100 msec を 3 回鳴らす）を確認する
		【説明】タイムアップ起動の実時間との確認は、TRF01_07 で確認する
	TSF01_15	タイマボタン 3 msec 以上長押し、リセットを確認する
		【説明】リセットはブザーを 100 msec 鳴らし、表示を 0（0 min0 sec）にする
	TSF01_16	時間設定範囲を確認する（最小から最大）、および最大超えを確認する
		【説明】最小：0 min0 sec、最大：60 min0 sec、最大超え：60 min0 sec から 1 min0 sec に表示遷移
	TSF01_17	タイマ表示に対して実測時間で確認する
		【説明】代表値で確認する、また秒の切り上げが行えているかも確認する

仕様を構文型に分類する際に足りない要素が推測できたときは補足してもらったうえで構文型に分類してもらう．補足できない場合は空欄（∅）としてもらう．R' が 25 文あるが，すべて R を補足する文である．表 9 の「【説明】59 min48 sec でタイマボタンを 1 回（100 msec）押したら、60 min0 sec をセットしたことになり、タイマ残り時間表示窓は 60 となる」が R' の例であり，その前の文に補足説明を加えている．そのため，これら R' を使っていなくても問題はない．

4.2 網羅性の比較方法

I_{tc} でのテスト項目は**表 11** のような項目に分類されている．対象とした要求仕様は機能を定義するものなので，提案手法により得られる項目は機能に関するテスト項目である．比較の対象は I_{tc} の機能テスト 56 項目とした．

テスト項目の比較は目視で行い，共通であるかどうかを判断する．表現が異なる場合でも，記述内容が同一の場合には，同一のテスト項目とする．たとえば，“タイマボタンを押すのを止めた 1 sec 後に起動することを確認する” と

表 11 テスト設計コンテストのテスト項目の分類と項目数
Table 11 The number of categories and test items in the test document.

	件数	割合
機能テスト	56	64%
その他	32	36%
合計	88	100%

表 12 テスト項目の比較結果
Table 12 Comparison of the results of the test items.

提案手法	I_{T_c}	共通テスト項目
58 項目	56 項目	50 項目

“タイマボタンを押すのを止めた 1 sec 後からタイマ起動するのを確認する” は同一のテスト項目であるとする．

4.3 結果

提案手法により，話題沸騰ポット要求仕様書第 7 版からテスト項目が得られた．テスト項目を得るまでに要した時間は 30 時間である．23 文の R から要求記述表 T_i を 23 件得た．今回の実験はすべて人手で行い自然言語処理ツール等は使用していない．R を構文型に分類する際に，要素の足りていない文があったため，8 文において，要素 8 件を補足した．補足できなかった要素は存在しなかった．たとえば，“タイムアップしたらブザーを 100 msec 間隔で 100 msec を 3 回鳴らす” は S_i である “システムは” を補足し，“タイムアップしたら（システムは）ブザーを 100 msec 間隔で 100 msec を 3 回鳴らす” とした．なお，“100 msec 間隔で 100 msec” は Nr，“3 回” は Nv である．

構文型はそれぞれ $CSONF$ 型 7 文，CFN 型 3 文，CNN 型 5 文，SCF 型 8 文であった．構文型の要素はそれぞれ S15 件，O7 件，C23 件，F18 件，N15 件であり．そのうち S3 件，C5 件は補足した．これらから要素を取り出し，ガイドワードを用いて 183 件の逸脱を得た．テスト項目 I_{T_i} は 23 項目，I_{D_i} は 35 項目得た．

I_{tc} との比較結果は**表 12** のとおりである．共通のテスト項目は 50 項目あった．提案手法で得たテスト項目は，I_{tc} のテスト項目の約 91%を含んでいた．提案手法で得られ，I_{tc} のテスト項目になかった項目は，すべて要求仕様書の記述漏れから作成したテスト項目である．たとえば，要求仕様 R_i “保温ランプ，沸騰ランプをともに消灯する” は暗黙の条件 C_i “温度制御停止中のときは” が明示されていなかったため，これを補足し “(温度制御停止中のとき）保温ランプ，沸騰ランプをともに消灯する” とすることで，“温度制御停止中以外のとき保温ランプ，沸騰ランプをともに消灯しないことを確認” というテスト項目を作成した．

I_{tc} のテスト項目に含まれ，提案手法により得たテスト項目に含まれなかった項目はすべて話題沸騰ポット要求仕様書第 7 版には記述されていない仕様をテストするものであった．具体的には，I_{T_c} のテスト項目にはタイマ表示の時間が正しい時間を表示するかどうかを確認する “タイマ表示に対して実測時間で確認する” がある．もし要求仕様書にタイマの主体前提条件として “カウントダウンは実測時間で行われる” があれば，逸脱 “カウントダウンは実測時間より早い” と “カウントダウンは実測時間より遅い” を得ることができたが，これはなかったため提案手法ではテスト項目として得られなかった．

5. 考察

提案手法により要求仕様書からテスト項目を作成することができた．テスト設計コンテストにおいて実務経験者が設計したテスト項目と比較した結果，本手法で得たテスト項目は I_{tc} のテスト項目の約 91%を網羅していた．学生の被験者により，実務者が作成したテスト項目の約 9 割を含んでいる点で，提案手法の有効性を示していると考える．

提案手法により得られたテスト項目のうち，I_{tc} に含まれなかったテスト項目が 8 項目あった．これらのテスト項目は要求仕様書に記述がない内容を確認するものであり，被験者が要求仕様書の文を分類する際に補足したものである．実験では構文型に分類する際に，要求仕様書の文に不足している要素がないかを確認した．“(温度制御停止中は）沸騰ランプ，保温ランプをともに点灯する” のようなポットを使用する環境をはじめとする前提条件が不足していた．提案手法では要求仕様書を 1 文 1 文チェックしていくことにより，こうした条件や制約の記述漏れの検出が期待できる．

一方，提案手法では得られず，I_{tc} には含まれていたテスト項目が 6 項目あった．I_{tc} の作成チームは要求仕様に対して 6W2H で分析 [1] を行い，関心事の洗い出しを行っており [7]，“タイマ表示と実際の時間が一致しているか確認する” といったテスト項目があげられていた．要求仕様書に “カウントダウンは実際の時間で行われる” といった文があれば提案手法でも得られるテスト項目であったが，要求仕様書にはそうした文が含まれなかったため，提案手法ではテスト項目として得られなかった．

今回用いた要求仕様書は USDM で記述され，複数回の改版を得て，極力曖昧な表現をなくして書かれた要求仕様書であり，記述されている文は要求構文型に分類することが容易であった．USDM においても要求仕様は自然言語で記述するため，一般的な自然言語で記述された要求仕様書と大きな差はないと考えられるが，一般的な要求仕様書が曖昧さの除去のために複数回改版されるとは限らない．要求文が明確に定義され，提案手法の手順 1 の実施において，補足の必要のない要求文が与えられれば提案手法の効果が得られるが，手順 1 の要求記述表が適切に作成できない場合には，効果が限定される．Myers の三角形問題 [5] の要求文をもとに，原文そのままで提案手法を試行した場

合と手順 1 の実施において実施者が要求文を補足する必要のない要求文を与えた場合とで提案手法の実施結果を比較してたものを付録に示している．一般的なソフトウェア開発で作成される要求仕様書においてどの程度補足が必要になるかを確かめ，提案手法を評価することは今後の重要な課題の 1 つである．

提案手法の各手順の自動化と属人性について考察する．3 手順とも完全に自動化することは現時点では難しい．しかし，手順 2 においてガイドワードと要求記述表の各要素を組み合わせて，逸脱の候補を作成することは可能である．候補の中には “2 秒以内より早い” のような逸脱として妥当かどうかを人手によって判断しなければ，手順 3 を実施できないものもあるためである．手順 3 は手順 2 と同様に要求記述表 T_i の各要素と逸脱から候補を作成できるが，テスト項目とするためには，不適切なものを人手によって区別する必要がある．これらが手順 2，3 を完全に自動化できない理由である．手順 1 は，手順 2，3 と同様に自然言語処理により主体 S や対象 O の候補を提示し，人手により適切なものを選択することはできる．しかし，要求文 R_i に明示されていないことの補足，他の要求文 R_j（$i \neq j$）からの補足が必要な場合には，人手以外の方法では実施が難しい．

6. おわりに

要求仕様書の記述を分析し，HAZOP を用いてテスト項目を得る手法を提案した．提案手法では，要求仕様書の文を IEEE29148 で定義されている 4 つの構文型に分類し，主体，対象，機能，条件，制約といった要素に分解する．得られた要素にガイドワードを適用し，逸脱を得る．得られた逸脱を再現するようなテスト項目を異常系のテスト項目とし，要求仕様書どおりのテスト項目（正常系のテスト項目）を加えてテスト項目を得る．

提案手法で実際にテスト項目が得られるか評価するために学生を被験者とする実験を実施した．要求仕様書は，教育用に作成され，公開されている “話題沸騰ポット要求仕様書第 7 版” とした．実験の結果，要求仕様書から 58 項目のテスト項目を得た．また，同要求仕様書を対象として，実務者が提案手法によらず作成した機能テストの項目 56 項目と比較したところ，50 項目が共通していた．比較対象とした実務者のテスト項目は特定非営利活動法人ソフトウェアテスト技術振興協会が実施したテスト設計コンテスト’13 において本戦に出場したチームが作成したテスト項目である．ほぼ同等のテスト項目を学生の被験者により得られたことは提案手法の有用性を示唆している．

提案手法では，要求仕様書が明確に記述されていることを前提にテスト項目を得ているが，要求仕様書が必ずしも明確に記述されている場合ばかりではない．明確に記述されていない場合でも，要求仕様書の記述を補完しながら，網羅的なテスト項目を得るために提案手法を改良することは今後の課題である．

参考文献

[1] 秋山浩一：HAYST 法によるテスト分析 テスト設計，ASTER 智美塾配布資料 (2012).

[2] ASTER：テスト設計コンテスト’13 本選決勝大会レポート (2013)，入手先 ⟨http://aster.or.jp/business/contest/contest2013.html⟩.

[3] Cohen, M.B.: Constructing strength three covering arrays with augmented annealing, *Discrete Mathematics*, Vol.308, Issue 13, pp.2709–2722 (2008).

[4] David, J.P.: The Principled Design of Computer System Safety Analyses, Ph.D. Thesis, The University of York (1999).

[5] Glenford, J.M.: *The Art Of Software Testing 3rd Edition*, Wiley (2011).

[6] Habli, I.M.: Model-based assurance of safety-critical product lines, Ph.D. Dissertation, Department of Computer Science, University of York (2009).

[7] HITACHI：話題沸騰ポット（GOMA-1015 型）テスト要求分析書 第一版 (2012)，入手先 ⟨http://www.hitachi.co.jp/Prod/comp/soft1/topics/jasst/02.pdf⟩.

[8] International Software Testing Qualifications Board: *Standard Glossary of Terms used in Software Testing Version 2.4*, p.29 (2014).

[9] International Software Testing Qualifications Board: *Advanced Level Syllabus Test Analyst Version 2012.J01* (2012).

[10] Jacek, C.: Pairwise Testing in the Real World: Practical Extensions to Test-Case Scenarios, *Proc. 24th Annual Pacific Northwest Software Quality Conference*, pp.419–430 (2006).

[11] 河野哲也：ソフトウェア要求仕様における HAZOP を応用したリスク項目設計法，ソフトウェアシンポジウム 2012 東京 (2012).

[12] Leveson, N.G.: Fault Tree Analysis, *Safeware: System Safety and Computers*, pp.317–326, Addison-Wesley (1995).

[13] Leveson, N.G.: Failure Mode and Effects Analysis, *Safeware: System Safety and Computers*, pp.341–344, Addison-Wesley (1995).

[14] Leveson, N.G.: Hazards and Operability Analysis, *Safeware: System Safety and Computers*, pp.335–341, Addison-Wesley (1995).

[15] McDermid, J.: Issues in developing software for safety critical systems, *Reliability Engineering and System Safety*, Vol.32, No.1–2, pp.1–24 (1991).

[16] Reifer, D.J.: Software Failure Modes and Effects Analysis, *IEEE Trans. Reliability*, Vol.R-28, No.3, pp.247–249 (1979).

[17] Reza, H., Ogaard, K. and Malge, K.: A model based testing technique to test web applications using statecharts, *Proc. 5th International Conference on Information Technology: New Generations*, pp.183–188 (2008).

[18] Robert, M.: Orthogonal Latin Squares: An DAapplication of Experiment Design to Compiler Testing, *Comm. ACM*, Vol.28, No.10, pp.1054–1058 (1985).

[19] Sarma, M. and Mall, R.: Automatic Test Case Generation from UML Models, *Proc. 10th International Conference on Information Technology*, pp.196–201 (2007).

[20] SESSAME：話題沸騰ポット（GOMA-1015 型）要求仕様書第 7 版 (2005).

[21] SESSAME：話題沸騰ポット（GOMA-1015 型）テストアーキテクチャ設計書 第一版 (2012)，入手先 ⟨http://www.hitachi.co.jp/Prod/comp/soft1/topics/jasst/03.pdf⟩.
[22] 清水吉男：要求を仕様化する技術・表現する技術 仕様が書けていますか?，技術評論社 (2005).
[23] Systems and software engineering – Life cycle processes – Requirements engineering, ISO/IEC/IEEE 29148:2011(E), pp.1–94 (2011).
[24] 山本修一郎：保証ケースに基づくテスト十分性に関する合意形成手法の提案，人工知能学会第 13 回知識流通ネットワーク研究会 (2013).

付　　録

Myers の三角形問題 [5] を対象として，提案手法を 3 名の実施者に試行してもらった．実施者 3 名は名古屋大学大学院情報科学研究科の大学院生である．試行は 2 回実施し，1 回目の試行では，提案手法の手順 1 において実施者が要求文を補足しながら必要な要素を抽出しなければならない要求文（**表 A·1**）を与えて提案手法を実施してもらった．2 回目の試行では，手順 1 において補足の必要がない要求文（**表 A·2**）を与えて提案手法を実施してもらった．

文献 [5] に記載されている正しいテスト項目（**表 A·3**）と試行の実施結果を比較したものを**表 A·4** に示す．表 A·4 の 1 行は 1 件のテスト項目に対応する．表 A·4 の ID は表 A·3 の ID に対応する．表 A·4 の手順 1，2，3 の列は手順 1，2，3 の実施結果を示しており，“○” は正解，“×” は不正解，“-” は対応する手順がないことを示す．ある手順で “×” となったものは，それよりも後の手順も不正解であるため空白としている．

表 A·4 のとおり，補足の必要のある要求文を与えた場合（1 回目の試行），手順 1 が不正解となり，それ以降の手順の回答が誤っていた．補足の必要のない要求文を与えた場合（2 回目の試行），手順 1 の結果はすべて正解となり，続く手順においても不正解は 3 件のみであった．不正解の原因はいずれも作業ミスである．これらの結果から，手順 1 が正しく実施できると以降の手順の誤りは少なく，手順 1 の実施にあたって要求文に補足が必要のない要求文が与えられると，提案手法の効果が得られることを示しているといえる．

表 A·1　例題文 [5] の分類
Table A·1　Categorized requirement specification statements in the article [5].

ID	要求文
R'	あるプログラムのテストを実施する．
R_1	このプログラムは、カードから 3 つの整数を読む．
R'	この 3 つの値は、それぞれ三角形の 3 辺の長さを表すものとする．
R_2	カードを読み取ったとき、プログラムは、三角形が不等辺三角形か二等辺三角形か正三角形を決めるメッセージを画面上に印字する．
R'	これらの条件から、プログラムのテストケースを考えよ．

表 A·2　例題文（補足版）の分類
Table A·2　Categorized requirement specification statements revised.

ID	要求文
R_{2-1}	プログラムは 3 辺の整数 (a, b, c) を入力し $a < b+c, b < a+c, c < a+b$ を全て満たすとき，$a = b = c$ ならば正三角形，$a = b$，$b = c$，$a = c$ のいずれかを満たすとき二等辺三角形，$a = b = c$，$a = b$，$b = c$，$a = c$ のいずれも満たさないとき不等辺三角形を印字する．

表 A·3　Myers の三角形問題の解答 [5]
Table A·3　Answers for weinberg-Myers triangle problem [5].

ID	解答
I_{T_1}	有効な不等辺三角形となる値のセット解答
I_{T_2}	有効な正三角形となる値のセット解答
I_{T_3}	有効な二等辺三角形となる値のセットで a = b の場合
I_{T_4}	有効な二等辺三角形となる値のセットで b = c の場合
I_{T_5}	有効な二等辺三角形となる値のセットで a = c の場合
I_{D_1}	長さが 0 の辺がある場合
I_{D_2}	長さが負の値の辺がある場合
I_{D_3}	整数でない辺がある場合
I_{D_4}	3 辺が 0 (無効な値で 3 辺が等しい場合を代表している)
I_{D_5}	2 辺の和がもう 1 辺と等しくなる値のセットで a = b + c の場合
I_{D_6}	2 辺の和がもう 1 辺と等しくなる値のセットで b = a + c の場合
I_{D_7}	2 辺の和がもう 1 辺と等しくなる値のセットで c = a + b の場合
I_{D_8}	2 辺の和がもう 1 辺より小さくなる値のセットで a > b + c の場合
I_{D_9}	2 辺の和がもう 1 辺より小さくなる値のセットで b > a + c の場合
$I_{D_{10}}$	2 辺の和がもう 1 辺より小さくなる値のセットで c > a + b の場合

表 A·4 結果
Table A·4 The result of the evaluation with weinberg-Myers triangle problem.

	1 回目									2 回目								
	実施者 A			実施者 B			実施者 C			実施者 A			実施者 B			実施者 C		
ID	手順 1	手順 2	手順 3	手順 1	手順 2	手順 3	手順 1	手順 2	手順 3	手順 1	手順 2	手順 3	手順 1	手順 2	手順 3	手順 1	手順 2	手順 3
I_{T_1}	×	-		×	-		○	-	○	○	-	○	○	-	○	○	-	○
I_{T_2}	×	-		×	-		○	-	○	○	-	○	○	-	○	○	-	○
I_{T_3}	×	-		×	-		×	-		○	-	○	○	-	○	○	-	○
I_{T_4}	×	-		×	-		×	-		○	-	○	○	-	○	○	-	○
I_{T_5}	×	-		×	-		×	-		○	-	○	○	-	○	○	-	○
I_{D_1}	○	×		×			×			○	○	○	○	○	○	○	×	
I_{D_2}	○	×		×			×			○	○	○	○	○	○	○	×	
I_{D_3}	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
I_{D_4}	○	×		×			×			○	×		○	○	○	○	○	○
I_{D_5}	×			×			×			○	○	○	○	○	○	○	○	○
I_{D_6}	×			×			×			○	○	○	○	○	○	○	○	○
I_{D_7}	×			×			×			○	○	○	○	○	○	○	○	○
I_{D_8}	×			×			×			○	○	○	○	○	○	○	○	○
I_{D_9}	×			×			×			○	○	○	○	○	○	○	○	○
$I_{D_{10}}$	×			×			×			○	○	○	○	○	○	○	○	○

大林 英晶

1991 年生．2014 年名古屋大学工学部電気電子情報工学科卒業．2014 年同大学大学院情報科学研究科情報システム学専攻入学．

森崎 修司 （正会員）

2001 年奈良先端科学技術大学院大学情報科学研究科博士課程修了．2013 年 10 月より名古屋大学大学院情報科学研究科准教授．ソフトウェアレビュー，エンピリカルソフトウェア工学の研究に従事．博士（工学）．IEEE，プロジェクトマネジメント学会各会員．

渥美 紀寿 （正会員）

2007 年名古屋大学大学院工学研究科情報工学専攻博士課程修了．2005 年南山大学数理情報学部情報通信学科講師，2010 年名古屋大学大学院情報科学研究科附属組込みシステム研究センター研究員，2011 年同センター特任助教，2012 年より同大学情報連携統括本部情報戦略室特任助教．博士（工学）．プログラム解析，ソフトウェア保守，ソフトウェア再利用等の研究に従事．日本ソフトウェア科学会，電子情報通信学会，IEEE，ACM 各会員．

山本 修一郎 （正会員）

1977 年名古屋工業大学情報工学科卒業．1979 年名古屋大学大学院工学研究科情報工学専攻修了．同年日本電信電話公社入社．2002 年（株）NTT データ技術開発本部副本部長．2007 年同社初代フェロー，システム科学研究所所長．2009 年東京工業大学統合研究院医療情報プロジェクト特任教授．同年名古屋大学情報連携統括本部情報戦略室教授．ソフトウェア工学，要求工学，IC カードプラットフォーム，ユビキタスコンピューティング，知識創造デザインの研究に従事．博士（工学）．電子情報通信学会，日本ソフトウェア学会，人工知能学会，日本情報経営学会，プロジェクトマネジメント学会，ACM，IEEE 各会員．