

研究論文

組込みシステムにおける熟練者知見を基にした
優先度付きテストケース自動生成手法の提案齋藤 英美^{1,a)} 中川 雄一郎¹

受付日 2020年2月25日, 採録日 2020年6月22日

概要: 複数機器を接続可能で機能も豊富である集中制御装置では、接続機器と機能の組合せが膨大となる。そのためテスト設計の熟練者は、自身の知見に基づいて検証する組合せを限定しテストケースを作成している。しかし知見は熟練者の暗黙知であるため、非熟練者との作業分担や自動化は困難であり、テストケースの作成に時間がかかる。本報告では、テスト設計者の熟練度を問わず、高品質なテストケースを短時間で作成可能にする手法を提案する。テスト熟練者が持つ知見を形式知化し、製品仕様や組合せ数など人により判断が変化しない情報を用いた優先度付与ルールに変換することで、優先度付テストケースを自動生成可能とした。提案手法を空調機の集中制御装置のテストに適用した結果、テスト設計者の熟練度によらず、不具合検出効率の良いテストケースを従来と比較して 1/10 程度の時間で生成できることを確認した。

キーワード: 組合せテスト, 優先度, テストケース生成, システムテスト, 空調

Proposal of Automatic Priority Test Case
Generation Method Based on Expert Knowledge in
Embedded SystemEMI SAITO^{1,a)} YUICHIRO NAKAGAWA¹

Received: February 25, 2020, Accepted: June 22, 2020

Abstract: The combination of the connected device and the function is large in the centralized control device which can connect multiple devices and is abundant in function. Therefore, the expert of the test design limits the combination which verifies based on the knowledge of oneself, and creates a test case. However, since the knowledge is implicit knowledge of the expert, it is difficult to share work, and the test design takes time. In this report, we propose a method that enables the creation of high quality test cases in a short period of time, regardless of the proficiency of the test designer. The test cases with priority are able to be automatically generated by standardizing knowledge of the test expert and converting into rules based on information that does not change the interpretation by the person. We apply the proposed method to the test of the centralized controller of the air conditioner. As a result, we confirm that a test case with a good defect detection efficiency can be generated in 1/10 of the time compared with the conventional, regardless of the proficiency of the test designer.

Keywords: combinatorial testing, priority, testcase generation, system test, air conditioning

1. はじめに

近年の組込みシステムでは、機器の多機能化や、IoT 化による接続機器の増加などで、検証する機能と接続機器の

組合せが増大している。組込みシステムでは、ソフトウェアを実際の機器上で動作させないと不具合が分からないことも多い。しかし機器の準備や機器上でのソフトウェア実行に時間がかかるため、すべての組合せをテストすることは困難である。そこで、少ないテスト件数で重大な不具合を検出することが組込みシステムにおいて重要になっている。テスト設計の熟練者は、自身の知見に基づき、重大な

¹ 株式会社日立製作所
Hitachi, Ltd., Yokohama, Kanagawa 244-0817, Japan
^{a)} emi.saito.tj@hitachi.com

不具合を検出可能である組合せを判断しテストケースを作成している場合が多い。しかし知見は熟練者の暗黙知であるため、非熟練者との作業分担や自動化は困難であり、テスト設計に時間がかかる。加えて熟練者間でも知見に差異があり、テストケースの内容にばらつきが出やすい。

少ないテスト件数で重大な不具合を検出可能にする方法の1つとして、テストケースに優先度を付与する手法が報告されている [1], [2], [3], [4]。既存研究の多くは回帰テストに対する手法であり、ソースコードの情報に基づき優先度を算出する方法 [1], [2] や、以前のテスト結果から優先度を決定する方法などが報告されている [3]。これらは、回帰テストを対象とするためテストケースが与えられている前提であり、新規開発案件への適用は難しい。ほかには、組合せテスト [5] に優先度を付与する手法 [4] があり、重みが設定された組合せテストの入力モデルからテストケースの生成と優先度付けをする。しかし重み付きの入力モデルは与えられている前提であり、モデルへ付与する重みを決めるには、熟練者の知見が必要になるとみられる。

そこで本報告では、テスト設計者の熟練度を問わず、高品質なテストケースを短時間で作成可能にする手法を提案する。テスト熟練者が持つ知見を形式知化し、製品仕様や組合せ数などの個人によって判断が変化しない情報で表現した優先度付与ルールに変換することで、製品仕様情報だけを用いて優先度付与テストケースを自動生成可能とする。優先度付与のみでなくテストケースの生成を同時に行うので、新規開発案件への適用も可能である。これまでに、接続機器が1つの場合における優先度付与手法を報告している [6] が、本報告では、複数機器接続時の優先度も付与できるよう改良を行い、手法を評価した。2章では、対象とする組込みシステムの想定と課題を述べる。3章では課題解決のアプローチを説明する。4章では、提案手法を説明する。5章では、提案手法の評価を行う。6章はまとめである。

2. テスト対象と課題

2.1 対象の想定

本報告では、空調の集中制御装置をシステムテストの対象と想定する。空調は室外機、室内機、リモコンで構成される。加えて、複数の室内機を一括管理するために、図1に示すように集中制御装置を用いることがある。集中制御装置は、図のように、接続されている機器に対して個別あるいは複数の機器に同時に温度調節や運転/停止などの操作を行うことができる。ビル、病院、ホテルなど大型施設では、空調を一括管理するために、集中制御装置を導入している場合が多い。

集中制御装置の機能の1つに、ユーザからの操作に応じて冷房運転、暖房運転などの運転を切り替えるための「運転モード」の選択機能がある。通常は、図1のように暖房

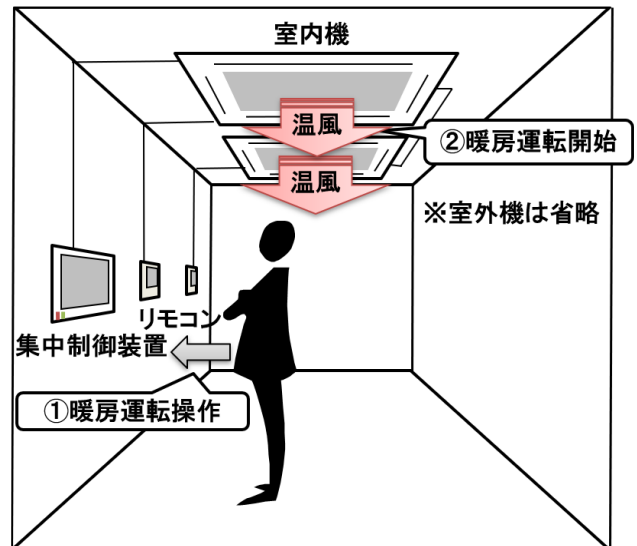


図1 集中制御装置の使用例

Fig. 1 Example of using central controller.

運転を選択すると、室内機は暖房運転を開始する。しかし、暖房が使用できない機器（ここでは冷専機と呼ぶ）が接続されている場合、集中制御装置は暖房運転の選択操作を受け付けられない動作する。また、暖房運転を受け付けられないためのソフトウェア的な設定（ここでは冷専固定と呼ぶ）を集中制御装置から指定した場合、接続されているのが冷専機であるかにかかわらず、集中制御装置は暖房運転を受け付けられないように動作する。以上から集中制御装置は、接続される機器や設定される機能が持つ制約に応じて、提供する機能の振舞いを変更する必要がある。

2.2 組合せの考え方と概算

集中制御装置では接続される機器や設定される機能が持つ制約に応じて、提供する機能の振舞いを変更することから、接続機器と設定機能の組合せをすべて検証することが理想である。しかし集中制御装置に接続可能な機器、設定可能なソフトウェア的な機能は多数存在し、集中制御装置が提供する機器操作の機能も多数存在するため、すべての組合せを網羅するとテストケースは膨大になる。

ここでテストケース数を見積もるため、集中制御装置が提供する機能1つにおいて、その機能に影響を与える機器および設定可能な関連機能の組合せ数を概算する方法を示す。集中制御装置のテスト対象機能に影響を与える接続機器の状態数を n とおく。接続機器の状態数 n は、機器の種類数 u と、機器に設定可能な関連機能数 f で表現できる。1つの関連機能は、有効（設定されている）と無効（設定されていない）の2状態を持つと見なすと、接続機器の状態数 n は式(1)のように表現できる。

$$n = u \times 2^f \tag{1}$$

集中制御装置に同時に接続する機器の数を r とおく。異

なる状態を持つ機器どうしを組合せて検証すると仮定すると、集中制御装置の最大接続数 $> n$ ならば、 r は、1 から n まで変化する。よって、組合せ数 N_{mul} は、式 (2), (3) を用いて計算できる。

$$N_{mul} = \sum_{r=1}^n {}_n C_r = {}_n C_1 + {}_n C_2 + \dots + {}_n C_n \quad (2)$$

$${}_n C_r = \frac{n!}{(n-r)!r!} \quad (3)$$

機器の状態数や組合せの考え方を図 2 に示す。空調の冷房、暖房などの運転を切り替える運転モードの選択機能がテスト対象の場合、テスト対象に影響を与える機器の例として冷専機がある。機器に設定可能な関連機能とは、前述の冷専固定設定や、冷房/暖房の自動的な切換え運転を許可する冷暖自動設定である。機器 1 つに対して状態数が n 個あり、それが接続機器数 r 分存在するときの組合せが N_{mul} である。

式を用いて、組合せを概算する。接続機器のうちテスト対象とする機能に影響のある機器の種類は 4 つ ($u = 4$)、関連機能は 3 つ ($f = 3$) と仮定した場合、式 (1) を使うと、 $n = 32$ となる。よって組合せ数 N_{mul} は式 (4) のように計算できる。

$$N_{mul} = \sum_{r=1}^{32} {}_{32} C_r = 4,294,967,295 \quad (4)$$

以上から、1 つの機能に対して影響のある機器と設定可能な関連機能に絞っても約 40 億通りの組合せが存在する。

なお、不具合の多くは 2 状態までの組合せで起きることが知られており [7]、 $r = 1 \sim 2$ に範囲を絞った場合は、 $N_{mul} = 528$ 通りの組合せに限定できる。それでも 1 機能に対して数百件以上の組合せであり、実際には多数の機能がテスト対象となることから、すべての組合せを実際の機器で検証することは困難である。

2.3 課題

組合せを概算した結果、すべての組合せを実際に動作検証することは困難であるとの見通しを得た。検証する組合せを絞り込む必要があるが、単純に組合せ数を減らすだけでは、不具合の見逃し要因となる恐れがある。このため実際にはテスト設計の熟練者が、自身の知見に基づいて重要な組合せをつと検討し、実際の機器で検証可能な件数に収まるようにテストケースを作成していることが多い。しかし熟練者の知見は暗黙知であるため、非熟練者との作業分担や自動化は困難であり、テストケース作成作業に時間を要する原因となっている。また、作業を担当する熟練者間でも知見にばらつきがあり、テストケースの内容に差が出る問題がある。この問題は、たとえば重大な不具合を検出する組合せが漏れる、あるいは同様の不具合しか検出しない組合せを多数採用するなどにより、テストケースの品質を低下させる可能性がある。

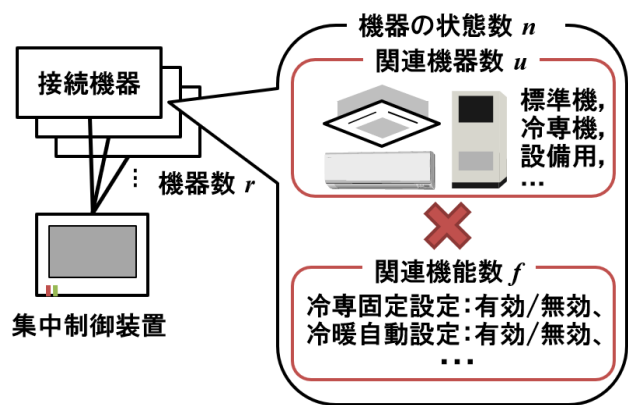


図 2 機器の状態と組合せの考え方

Fig. 2 Image of combination and the state of unit.

よって、現状の熟練者によるテストケース作成作業では、以下の課題が存在する。

- 課題 (1) テストケースの作成に時間がかかる。
- 課題 (2) テストケースの内容 (重大な不具合を検出する組合せの判断) にばらつきが出る。

3. 解決アプローチ

3.1 解決方針

前述の課題は、テストケース作成作業が熟練者に依存していることが原因である。そこでテスト設計の熟練者の暗黙知となっている知見を形式知化し、テスト設計者の熟練度に寄らずに利用可能な手法へと変換することで、短時間で高品質かつ誰でも同様のテストケースを作成可能にする。

少ない件数で重大な不具合を検出可能なら、不具合検出効率の良い高品質なテストケースと見なせる。検証する機能に対する接続機器と関連機能の組合せを生成し、重大な不具合を検出可能な組合せには高い優先度を付与することで、高品質なテストケースを実現することが考えられる。

組合せを生成し優先度を付与するテストケースを生成する手法として、重みを設定した入力モデルから優先度付きの組合せテストを自動生成する方法も報告されている [4]。しかし重みの設定基準に言及はなく、人によって判断が変化しテストケースの優先度が変わる可能性がある。

誰でも同様のテストケースを生成するためには、テスト設計者による判断が必要な作業を減らして作業結果の差分をなくすことが必要である。人による判断を減らせば、作業は手順化され自動化が容易であり、短時間化も実現できる。そこで本報告では、優先度付きテストケースを自動生成する手法を新規に構築する。

3.2 手法の要件

解決方針に沿った手法を構築するために、手法が満たすべき要件を以下に示す。

要件 (1) 熟練者の知見を形式知化し手法に反映する。

テストの組合せを絞り込む組合せテスト手法の研究は多

数あるが [4], [7], [8], これらの手法は, どの組合せも不具合の検出率あるいは検出する不具合の重要性は同じであるとの前提条件に基づいている. しかし, 実際には組合せの重要性には差がある場合がある. テスト設計の熟練者は, 重大な不具合を検出する組合せについて知見を持っており, その知見に基づいてテストケースを作成することで, 組合せ数を限定しても不具合の見逃しを抑制している. 熟練者の知見を形式知化して手法に取り入れることで, 熟練者以外が作業を行っても, テストケースの品質を保つことを可能にする.

要件 (2) 期待結果を含んだテストケースを生成する.

テストケースを生成する手法は多数存在するが, それらの多くは期待結果の生成を対象としていない. テストにおいて期待される動作結果は製品仕様に依存するため, 手法などで自動生成することは困難とされる. このため期待結果の作成は, テスト設計作業の短時間化を妨げる要因の 1 つである. 提案手法では, 人の判断の介入を減らすために, 期待結果も自動生成の対象とする.

3.3 要件の実現方針

3.3.1 要件 (1) の実現

熟練者の知見を明らかにする方法として, 熟練者自身へのヒアリングがある. しかし, 知見は暗黙知であり, 熟練者自身でも明確に表現することが難しい. 加えて, 熟練者間での知見のばらつきがあることが考えられる. そのため, 単なるヒアリングによって知見を形式知化することは難しい. ほかに, 機器と機能の組合せを生成し, 過去のテストケースとの差分をとることで, 熟練者が重要視する組合せを明らかにし, 熟練者の知見を推測する方法がある. しかし, 熟練者の知見のばらつきは熟練者の作成したテストケースにも引き継がれるため, 過去のテストケースだけを根拠にしても課題 (1), (2) を解決できない.

そこで, 次の手順に従うことで, ヒアリング, 過去のテストケース活用時のそれぞれの問題点を解消し, 知見の抽出と形式知化を行う.

まず, 接続機器と関連機能の組合せを作成して過去のテストケースとの差分をとり, 過去のテストケースに存在する/しない組合せを明らかにする. 過去のテストケースとの差分に対し発生理由の仮説を立て, 複数の熟練者に意見をいただいて検証を行う. 具体的な事例をもとにしたヒアリングとなるため, 抽象的な知見を具体化し引き出しやすくする. 加えて, 過去のテストケースに誤りがあった場合に, 複数の熟練者から指摘をもらうことができるため, 過去のテストケースにおける熟練者の知見のばらつきを解消することができる.

次に, 複数の熟練者からの意見を集約し, テストのポリシーとして共通的な概念として表現する. 具体的には, 明らかにした各人の個別の差分発生理由に対し, 特定の機能

に依存しない情報で表現できるような共通項を見つける. 共通項どうしを上位の概念でまとめなおし, テスト設計のポリシーとして定義する. 意見の集約と上位概念化作業により, ヒアリング対象の熟練者間の知見のずれをなくすることができる.

最後に, ポリシーを基にして, 機器と機能の組合せに対して優先度 (高・中・低など) を付与する. 複数の熟練者に対し, 優先度の判断結果が妥当か確認を行う. 優先度の問題があると熟練者が判断した場合, ポリシーを再検討する.

3.3.2 要件 (2) の実現

テストケースは, 事前条件, イベント, 期待結果で構成する. 本報告において, 事前条件は, 接続する機器や関連機能の組合せである. イベントは, 期待する結果を得るために機器へ行う入力操作をさす. たとえば, 暖房から冷房へ運転を切り替えるために, 運転モード画面を開いて運転モードを選択する操作である. 本報告ではイベントは自明なものとして, 手法による生成対象から除外する. 期待結果は, イベントに対して期待される機器の動作をさし, 本報告においては, テスト対象機能を操作した際に動作可能な範囲と定義する. この場合に期待結果は, 機器や設定が持つ動作範囲への制約の組合せで決められる. 制約の組合せに対するルールを定めることで, 期待結果をテスト設計の熟練者の判断なしに生成可能にする.

たとえば, 集中制御装置に標準的な室内機が接続されているとき, 運転モードの選択操作を行うと, 運転モードでは冷房, 暖房, ドライ, 送風が選択できる. このとき運転モードの動作範囲である期待結果は冷房, 暖房, ドライ, 送風である. 集中制御装置に冷専機が接続されているときには, 暖房を除外する制約が追加されるため, 運転モードの選択操作に対する期待結果は冷房, ドライ, 送風である. 冷暖自動の設定を行う関連機能が有効となっている場合, 冷暖自動運転を動作範囲に追加する制約が発生し, 期待結果は冷房, 暖房, ドライ, 送風, 冷暖自動である. しかし冷暖自動設定が有効でも, 冷専機が接続されている場合には冷房と暖房を使う冷暖自動運転は使用できないため, 期待結果は冷房, ドライ, 送風のままである. この動作仕様から, 冷専機による暖房禁止の制約は, 冷暖自動運転禁止の制約も含んでいると解釈できる. 加えて, 動作範囲から除外する制約は動作範囲に追加する制約よりも強いと解釈できる. このような法則性は特定機能ごとではなく製品に共通すると考えられる. 制約とその法則性を見出し, 期待結果生成のルールとして手法に取り入れることで, 自動的に期待結果を生成可能とする.

4. 提案手法の構築

4.1 熟練者のテスト設計ポリシー

集中制御装置の代表的な機能に対する過去のテストケースの分析と, 要件 (1) 実現時の熟練者ヒアリングから, 集

集中制御装置への接続機器が1つの場合と複数の場合で作成するテストケースを区別していることが分かった。本報告では、テスト対象である集中制御装置への接続機器が1つの場合のテストケースを単数機器テストケース、接続機器が複数の場合のテストケースを複数機器テストケースと呼んで区別することとする。

単数機器、複数機器のテストケースに対し、要件(1)の知見抽出手順に従うことで、熟練者の知見から以下のポリシーを明らかにした。

● 単数機器テストケースにおけるポリシー

(A) 利用頻度が高い利用形態におけるテストを優先する。
 (B) (A)に該当しない場合でも、不具合発生可能性が高いテストは優先する。

● 複数機器テストケースにおけるポリシー

単数機器テストケースのポリシーを踏襲するが、さらに以下のポリシーを追加する。

(C) 単数機器テストケースで確認が困難である内容を優先する。

(A)は、よく利用される形態での不具合は、利用者への影響が大きく、重大な不具合であるとの考えに基づいている。利用形態とは、集中制御装置に接続する機器やその機器に設定されている機能の組合せで表現できる[6]。たとえば、接続機器が標準機種で、特殊な設定を付けていない状態が最も利用頻度が高い利用形態である。

(B)には、矛盾する制約の組合せが該当する。冷専機では暖房が利用できないため、冷房と暖房を切り替える冷暖自動を設定するケースは存在せず、(A)のポリシーでは非優先と判断される。しかし、実際に冷専機に冷暖自動を設定してしまった場合、互いの動作が矛盾するので、不具合にならないか確認する必要がある。

(C)は、たとえば単数機器のテストケースでは確認ができない、異なる機器どうしの組合せを優先することをさす。

4.2 提案手法の概要

要件とその実現方針を反映した手法として、図3に示す優先度付きテストケース生成手法を提案する。

テスト条件は、製品の仕様情報を、因子と水準、制約などの要素に変換したものである。組合せテスト[5]の分野では、組合せの対象を因子と水準で表現し、異なる因子間の水準を組み合わせる。そこで提案手法でも、因子と水準で組合せの対象を表現する。なおここでの製品の仕様情報とは、たとえば空調の集中制御装置の運転モード機能において、「標準機種が接続されているとき、冷房、暖房、送風、ドライが選択できる。冷専機が接続されているとき、暖房は使用できない」といった内容である。

提案手法では、単数機器テストケースを優先度付与で絞り込み、高い優先度を持つ単数機器テストケースどうしを組み合わせることで複数機器テストケースを生成する。組

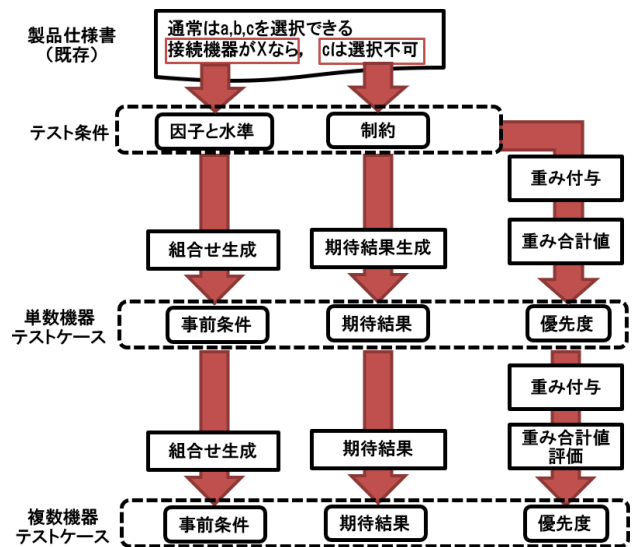


図3 優先度付テストケース生成手法概要
 Fig. 3 Overview of prioritized test case generation method.

テスト条件表				テスト対象				
				運転モード				
				範囲				
				冷房	ドライ	送風	暖房	冷暖自動
				基準範囲				
				○	○	○	○	
#	分類	因子	水準	制約				
1	F	冷専機	TRUE				▲	▲
not 1			FALSE					
2	P	冷暖自動	TRUE					◎
not 2			FALSE					
3	P	冷専固定	TRUE				▲	▲
not 3			FALSE					

図4 テスト条件
 Fig. 4 Test condition.

合せ数の増大を防ぐ方法の1つとして、組み合わせるパラメータを限定して組合せを生成し、生成結果と残りのパラメータを組合せて再度組合せを生成する手法が報告されており[8]、本手法でも、2段階の絞り込みを実施することで、複数機器テストケースの生成件数の抑制を可能としている。

製品仕様書からテスト条件を定義する部分のみ手作業を必要とし、単数機器テストケース、複数機器テストケースの生成は自動化可能とする。手法では優先度付与の方式としてルールによる重み付けを用いる。特定のルールに該当するごとに重みを付与し、その重み合計値を評価することで優先度を決定する。ルールには、明らかにした各テストケースのポリシーを反映する。

4.3 提案手法の詳細

図4に示す手法のそれぞれの要素について説明を行う。

4.3.1 テスト条件

テスト条件の定義は手作業で行う。しかし図4に示すよ

うなフォーマットにのっすることで、人による差分の発生を防ぎ、作業時間も短縮可能とする。

テスト条件には以下を含む。

(a) テスト対象

テスト対象の機能名である。

(b) 範囲

テスト対象機能の最大動作範囲を記載する。図の例では冷房、ドライ、送風、暖房、冷暖自動である。

(c) 基準範囲

すべての因子の水準が FALSE である場合の動作範囲を基準として記載する。すべての因子の水準が FALSE である場合は、標準的な室内機のみが接続され、関連機能がすべて無効の状態となる。その場合の動作範囲に該当する欄に「○」などの記号を記載する。

(d) 分類

機種に該当する因子には F, 事前に機器に設定しておく関連機能の因子は P, 運転中に変更可能な関連機能の因子は V に分類する。冷専機などの機種は F, 冷専固定, 冷暖自動などの事前に機種に設定する関連機能は P, 風向や風量など、運転中にも集中制御装置から変更可能な関連機能は V に分類する。

(e) 因子と水準

因子は、テスト対象機能の動作を制約する要因の名称とする。たとえば、冷専機などの機器名, 冷専固定などの関連機能名である。水準は FALSE, TRUE の 2 値で表現する。TRUE はその因子が有効, すなわち設定されている/接続されている状態である。冷専機が TRUE の場合, 集中制御装置の接続機器が冷専機である状態をさす。

(f) 制約

各因子に対して、水準が TRUE あるいは FALSE のときにテスト対象機能の動作範囲に与える影響を定義する。「▲」を記載した欄は動作範囲から除外する、「◎」を記載した欄は動作範囲に追加するという意味を持つ。たとえば因子が冷専機で水準が TRUE なら, 暖房運転および冷暖自動運転が利用できないため, 範囲のうち暖房と冷暖自動の欄に「▲」を記載して範囲からの除外を定義する。

4.3.2 単数機器テストケース

単数機器テストケースは図 5 に示す形態で定義する。本テストケースは、テスト条件に対し、組合せ生成, 期待結果生成, 重み付与, 重み合計値評価を実施して生成する。

(1) 組合せ生成

単数機器のテストケース生成では、異なる因子間で水準の組合せを網羅し、事前条件を生成する。なお、単数機器テストケースでは同時に複数の機器を接続することは対象外である。分類 F である因子が同時に TRUE となる組合せは禁則とし、組合せから除外する。

(2) 期待結果生成

制約を基に、因子間の水準組合せに対する期待結果を生

	事前条件			期待結果					重み	優先度
	#1	#2	#3	冷房	ドライ	送風	暖房	冷暖自動		
	F	P	P							
1	FALSE	FALSE	FALSE	●	●	●	●		0	高
2	FALSE	FALSE	TRUE	●	●	●			-2	高
3	FALSE	TRUE	FALSE	●	●	●	●	●	-2	高
4	FALSE	TRUE	TRUE	●	●	●			-5	中
5	TRUE	FALSE	FALSE	●	●	●			-2	高
6	TRUE	FALSE	TRUE	●	●	●			-6	低
7	TRUE	TRUE	FALSE	●	●	●			-5	中
8	TRUE	TRUE	TRUE	●	●	●			-9	低

図 5 単数機器テストケース

Fig. 5 Single unit test cases.

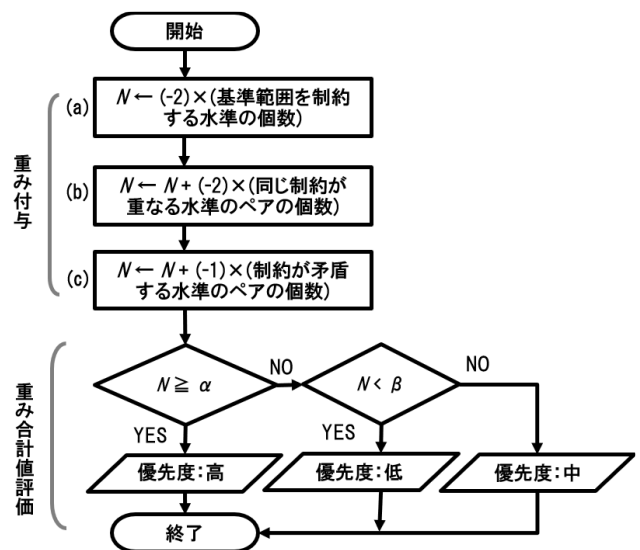


図 6 単数機器テストケースへの優先度付与方式

Fig. 6 Method to add priority to single unit test cases.

成する。期待結果の生成は、制約と基準範囲を組合せ、▲に該当する範囲は基準範囲から除外し、◎に該当する範囲は基準範囲に追加することで実現する。異なる制約が重なる場合には、制約の優位性に従って優先する制約を決める。制約は▲と◎の 2 種類で表現しているため、▲と◎が重なった場合にどちらを優先するか、制約の優位性を製品仕様を基に定義しておく。本報告で対象としている集中制御装置では範囲を除外する制約のほうが追加よりも強いため、▲と◎が重なる範囲は▲を優先するよう規定している。

(3) 重み付与と合計値評価

単数機器テストケースの優先度付与方法を説明する。図 6 に優先度付与のフローチャートを示す。図の N は重みの合計値である。(a)~(c) のルールに該当する組合せの重みを減らしていく仕組みとし、重みの合計値を評価して優先度を付与する。テスト熟練者の知見から抽出したポリシー (A), (B) を反映した重み付与ルールについて、以下に示す。

(a) 標準からずれるほど重みを減らす。具体的には、事前条件に True の数が多いほど重みを減らす。

(b) 組合せに同じ制約が重複すると重みを減らす。たと

例えば組合せる制約の同じ範囲に▲がある場合に該当する。

(c) 矛盾する制約の数だけ重みを減らす。たとえば、組み合わせる制約どうして同じ範囲に▲と◎がある場合に該当する。

重み付与ルール (a)~(c) は単数機器テストケースにおけるポリシー (A) を反映している。空調を利用する場合、基準範囲に対し制約がない状態、つまり標準機種を接続し関連機能の設定を無効にした形態での利用頻度が最も高い。逆に、冷専機を接続し冷暖自動設定と冷専固定設定を有効にする形態を利用することはほとんどない。後者の状態は、基準範囲を制約する水準を多数組み合わせた（水準が多数同時に TRUE になる）状態であり、ルール (a) によって重みが減らされる。加えて、冷専機と冷専固定は同じ制約であり、同時に有効にする意味がなく、利用頻度は低いため、ルール (b) によって重みが減らされる。さらに、冷専機と冷暖自動設定、冷専固定設定と冷暖自動設定はそれぞれ互いに矛盾する制約の組合せであり、利用頻度は低いため、ルール (c) によって重みが減らされる。以上により、利用頻度が低い利用形態の重みが小さくなるようにしている。

また、重み付与ルール (b), (c) はポリシー (B) も反映している。同じ制約の組合せと、矛盾する制約の組合せは、どちらも前述のように利用頻度は低いため、ポリシー (A) では非優先である。しかし矛盾する制約のほうが不具合可能性は高くなるため、ルール (b) での重みの減らし方を大きく、ルール (c) での重みの減らし方を小さくすることで、ポリシー (B) を反映している。

優先度は、重みの合計値があらかじめ定義した閾値を超えたかどうかで評価する。閾値は α と β で表現する。重みの合計値が α 以上なら優先度を高、 β 未満なら優先度を低、 α 未満で β 以上なら優先度を中と評価する。

本手法では、 α の値は -2 で固定とする。複数熟練者ヒアリング（要件 (1) の実現手順実施時）において、すべての熟練者が必ず実施する必要があると判断していたのは、「基準範囲を制約する水準が1つ以下（TRUE が1個か、すべて FALSE となる組合せ）」のテストケースである。このケースは、重みが必ず -2 以上になるため、 $\alpha = -2$ とおくことで、該当するテストケースを優先度“高”と判定可能である。 β は -5 とおく。重み合計値が -5 未満となるのは、基準範囲を制約する水準が3つ以上 TRUE となるテストケースすべてと、同じ制約を示す水準のペアが1つ以上存在するテストケースすべてである。制約を多数同時に組み合わせるのは利用頻度が低い形態であり、かつ組合せテストにおいて不具合の多くは2状態までの組合せで起きる [7] という知見からも優先度“低”に該当すると判定している。加えて、基準範囲を制約する水準が2つだけ TRUE であっても、その2つの水準が同じ制約を示す場合、利用形態が低く、不具合可能性も低いので、優先度“低”としている。基準範囲を制約する水準が2つ TRUE であり、かつ

	事前条件						期待結果				優先度	
	機器1			機器2			冷房	ドライ	送風	暖房		冷暖自動
	冷専機	冷暖自動	冷専固定	冷専機	冷暖自動	冷専固定						
1	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	●	●	●	●		0 高
2	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	●	●	●			0 高
3	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	●	●	●			-2 低

図 7 複数機器テストケース

Fig. 7 Multiple unit test cases.

互いに異なる制約を示し矛盾もしない形態（例：ドライ運転を禁止する制約と暖房運転を禁止する制約の組合せ）は、利用頻度は高く、重み合計値は -4 であり優先度“中”となるようにしている。基準範囲を制約する水準が2つ TRUE であり、かつ互いに矛盾する制約の場合は、重み合計値は -5 であり、優先度“中”である。ただしポリシー (A) を重視し利用頻度が低いという観点で、矛盾する制約の組合せが1つでもあれば優先度が低いと見なす熟練者もいることから、 $\beta = -5$ は基準値とし、テスト期間の余裕、テスト対象機能の仕様や重要性により β の値を調整してもよい。なお優先度を3段階で表現したのは、優先度“高”を最優先で実施して重大な不具合の有無を調査し、次に優先度“中”のテストを実施してより細かい不具合調査を行うといった段階的なテストへの活用を想定している。

4.3.3 複数機器テストケース

複数機器テストケースは図 7 に示す形態で定義する。本テストケースは、単数機器テストケースに対して、組合せ生成、期待結果生成、重み付与、重み合計値評価を実施して生成する。

(1) 組合せ生成

抽出した熟練者の知見をもとに、以下の手順によって組合せ数を絞り込むことを実現した。

Step (1) 単数機器の優先度“高”（重み合計値 α 以上）のテストケースを選出する。

Step (2) 単数機器テストケースで期待結果が同じであるテストケースは同値クラスとしてグルーピングする。同値クラスの中から、最も重みが大きいテストケースを代表のテストケースとして選出する。重みが大きいテストケースが複数あれば、事前条件の分類 F の因子が TRUE になっているテストケースを選出する。それでも複数存在する場合は、複数の代表テストケースを選出してよい。

Step (3) 異なる同値クラスから選定した代表テストケースを2つずつ組み合わせ、テストの事前条件を生成する。これは、ポリシーのほか、不具合の多くは2状態までの組み合わせで起きることが知られていることを根拠に [7]、同時に組み合わせるテストケースを2つとしている。

上記の手順は、複数機器テストケースのポリシー (C) を反映している。

(2) 期待結果生成

複数の接続機器を集中制御装置から操作する場合、複数の接続機器において共通する動作の範囲、あるいは各接続機器の動作範囲を合算した範囲のどちらかを選択可能とす

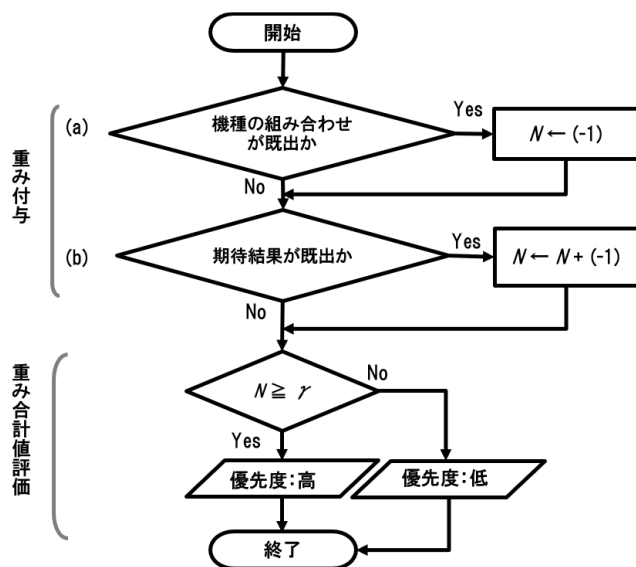


図 8 複数機器テストケースへの優先度付与方式

Fig. 8 Method to add priority to multiple unit test cases.

ることが多い。本報告では、複数機器に共通する動作の範囲だけを選択可能であると想定し、組み合わせる単数機器テストケースの期待結果どうしの AND をとることで、複数機器テストケースの期待結果を導出している。もし複数機器の動作範囲を合算した範囲を選択可能にする場合には、組み合わせる単数機器テストケースの期待結果どうしの OR をとることで、期待結果を生成可能である。どちらも期待結果生成の難しさに差異はないため、手法を実際にテストケース自動生成ツールとして実装する場合には、両方の期待結果生成手段を実装し選択可能にするとよい。

(3) 重み付与と合計値評価

優先度付与に関するフローチャートを図 8 に示す。

重みは、テストケース生成順に以下のように付与する。

(a) 期待結果が既出なら重みに -1 を付与する。

(b) 事前条件の機種どうしの組み合わせが既出なら重みに -1 を付与する。

(a) と (b) は、複数機器テストケースの生成順に優先度を付与していき、該当テストケースよりも前に生成した複数機器テストケースにおいて、該当テストケースと同じ期待結果、機種の組み合わせが出現しているかどうかで重みを減らす。上記の重み付与ルールは、複数機器テストケースのポリシー (C) を反映している。

重み合計値が γ 以上のときに優先度 “高”, γ 未満を “低” と評価する。複数機器のテストケース、組み合わせ生成時点で十分に生成されるテストケース数が絞り込まれているため、優先度 “中” の該当判定は行わず, “高”, “低” の 2 種類で優先度を表現する。本手法では、重み合計値は 0 から -2 までの範囲でしか変化せず、中間値をとって $\gamma = -1$ とする。

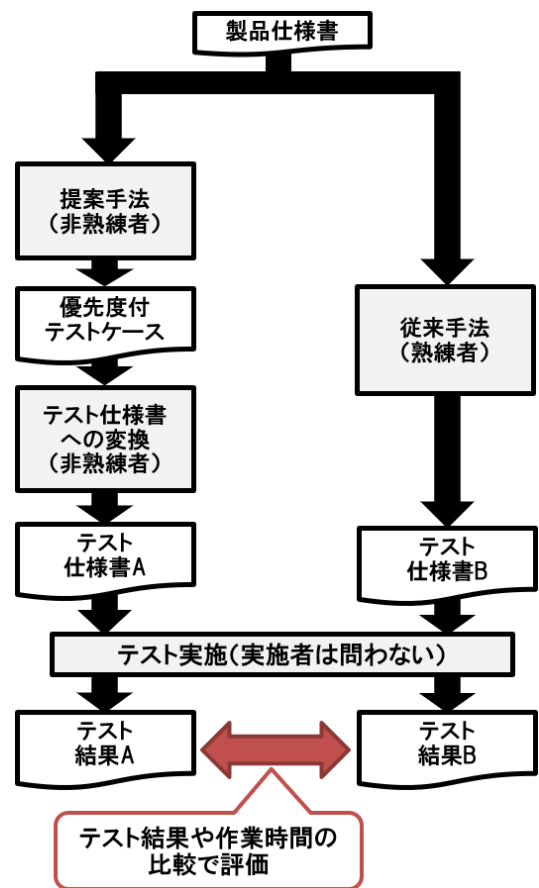


図 9 手法評価実験

Fig. 9 Evaluation experiment of the proposed method.

5. 評価

5.1 手法の評価方法

実現した手法を、新規の集中制御装置開発におけるシステムテストの一部に適用し、提案手法の効果を確認する。評価方法を図 9 に示す。

提案手法と従来手法のそれぞれを用いてシステムテストを作成する。テストケース作成に要する時間と、検出した不具合の比較で提案手法を評価する。従来手法とは、製品仕様書を基に、テスト設計の熟練者が直接テスト仕様書を作成する方法である。提案手法は、非熟練者が利用する。なお、従来手法との作業時間の比較に公平性を持たせるため、提案手法では、生成したテストケースをテスト仕様書の形態に手作業で記載しなおす作業も行った。ここでのテスト仕様書とは、テストケースとテストの実行手順が併記されたものである。

提案手法は、Excel VBA を用いてツールとして実装し利用した。Excel 表形式のテスト条件表へ仕様情報を入力することで、自動で優先度付きの単数機器テストケース、複数機器テストケースを生成可能である。テスト条件表の作成に関しても、短時間化の実現に向けた入力補完・支援機能を搭載している。

表 1 提案手法による生成テストケース数

Table 1 The number of generated test cases.

優先度	単数機器テスト ケース [件]	複数機器テスト ケース [件]
高	7	6
中	8	-
低	17	9

表 2 テスト作成時間の比較

Table 2 Comparison of the test design time.

提案手法 [h]	テスト仕様書への 変換 [h]	従来手法 [h]
0.5	5	78

5.2 評価結果

重み合計値の評価基準を $\alpha = -2$, $\beta = -5$, $\gamma = -1$ とし、実装した提案手法を用いて、テストケースを生成した。生成結果を表 1 に示す。

ここでテスト対象機能は、接続機器の種類数 $u = 4$ 、関連機能数 $f = 3$ であり、式 (1) に基づくと接続機器の状態数 $n = 32$ である。単数機器テストケースは、同時に接続する機器数 $r = 1$ の状態に相当するので、存在する組合せ数は、式 (4) に基づくと、 $N_{mul} = 32$ 件となる。複数機器テストケースは、組合せ生成時に 2 状態の組合せに限定しており、 $r = 2$ であるため、式 (4) に基づくと $N_{mul} = 496$ 通りの組合せが存在する。

提案手法において、単数機器テストケースは、生成する件数は減らさずに優先度を 3 段階で付与している。単数機器テストケースは全 32 件生成され、優先度“高”が 7 件、“中”が 8 件であった。本来の件数に対し、優先度“高”に着目すると約 1/5、“高”と“中”までを見ると約 1/2 にまでテストケース数を削減できている。複数機器テストケースは件数が多くなりやすいため、優先度を付与する前に、組合せ生成時点で生成数を減らす仕組みにしている (4.3.3 項 (1) 参照)。手法で生成数を減らしたことで組合せは全 15 件生成され、優先度“高”が 6 件であった。本来生成される $N_{mul} = 496$ に対して、組合せ生成数の限定と優先度付与によってテストケース数を約 1/100 に削減している。以上の結果から、提案手法によって、実施すべきテストケース数を十分に少ない数へ絞り込むことができている。

次に、提案手法と従来手法によるテスト仕様書作成時間を計測した結果の比較を表 2 に示す。

提案手法によるテストケース生成時間は 0.5 時間であった。生成したテストケースを用いてテスト仕様書を作成する作業には、その 10 倍の時間を要した。テスト仕様書でテスト手順記載方法の検討に時間を要したことが原因であった。一方で、従来手法は製品仕様書から直接テスト仕様書を作成しており、78 時間を要している。熟練者がテスト

表 3 不具合検出数

Table 3 Number of defects detected.

利用手法	不具合検出数 [件]
提案手法	1
従来手法	0

ケースを考えながら直接テスト仕様書を作成している場合と比べ、提案手法の活用により、テスト仕様書への手作業での変換を含めても 1/10 以下にまで作業時間を削減できることが分かった。よって、テストケース作成作業の課題 (1) を提案手法で解決できている。

次に、提案手法と従来手法によって作成したテスト仕様書における不具合検出数の比較結果を表 3 に示す。なお提案手法では、不具合の検出漏れ有無の確認のために、優先度にかかわらず生成したすべてのテストケースについてテストを実施している。

従来手法では不具合検出数が 0 件なのに対し、提案手法では 1 件不具合を検出した。不具合を検出したのは、優先度“中”の単数機器テストケースであった。検出した不具合は、利用者への影響度としては小さいものであった。重大な不具合を検出する組合せを優先度“高”、重大ではないが不具合を検出しやすい組合せを優先度“中”になるように手法を構築しており、狙いどおりに優先度付与できていることが分かった。従来手法および提案手法で生成されたテストケースの比較を行うと、従来手法で作られたテストケースのほうが組合せの網羅率が低いことが分かった。不具合を検出できなかった点からも、従来手法では、熟練者の知見の偏りや考慮もれがそのままテストケースの内容に反映されてしまったといえる。提案手法では、過去の熟練者の知見を集約・平均化して適用可能にしたことで、作業者の知見の不足によるテストケース生成結果のばらつきを解消でき、誰でも同一のテストケースを生成可能である。以上からテストケース作成作業の課題 (2) を解決できる。

6. おわりに

テスト設計の熟練者の知見を反映した優先度付きテストケース生成手法を提案した。提案手法を空調の集中制御装置のシステムテストに適用し、従来と比較して短時間で不具合検出効率の良いテストケースを生成できることを確認した。提案手法は空調の集中制御装置を対象としたが、抽出した知見や提案した手法、あるいは知見の抽出方法などは、他の組込みシステムでも有効であると考えている。手法の汎化、再利用性の向上についても検討を進める。

参考文献

- [1] Rothermel, G., Untch, R.H., Chu, C. and Harrold, M.J.: Prioritizing Test Cases For Regression Testing, *IEEE Trans. Software Eng.*, Vol.27, No.10 (2001).

- [2] 加藤正恭, 小川秀人: テスト戦略のためのテストケース優先度付けと可視化手法, 信学技報, Vol.117, No.136, pp.151-156 (2017).
- [3] Qu, B., Nie, C., Xu, B., Zhang, X.: Test Case Prioritization for Black Box Testing, *Proc. 31st Annual IEEE International Comput* (2007).
- [4] Choi, E., Kitamura, T., Artho, C. and Oiwa, Y.: Design of Prioritized N-Wise Testing, *Proc. International Conf. Testing Software and Syst.*, pp.186-191, Madrid (2014).
- [5] 秋山浩一: ソフトウェアテストの最新動向 3. 組合せテストの作成, 情報処理, Vol.49, No.2 (2008).
- [6] 斎藤英美, 利國 愛, 中川雄一郎: システムテストに対するユーザの利用形態に基づく優先度付きテストケース生成手法の提案, 信学技法, Vol.117, No.477, pp.69-74 (2018).
- [7] 土屋達弘, 菊野 亨: ペアワイズテスト—ソフトウェアテストの効率化を求めて, 電子情報通信学会論文誌 D, Vol.J90-D, No.10, pp.2663-2674 (2007).
- [8] Ozcan, M.: Applications of Practical Combinatorial Testing Methods at Siemens Industry Inc., *IWCT 2017* (2017).



斎藤 英美

2013年北海道大学大学院情報科学研究科修士課程修了。同年株式会社日立製作所入社。組込みシステムの生産性向上やスマートホームに関する研究に従事。



中川 雄一郎

1999年東海大学工学部航空宇宙学科卒業。2001年同大学大学院工学研究科航空宇宙額学専攻修士課程修了。2005年株式会社日立製作所入社。ソフトウェアテスト技術, 形式手法等の研究開発に従事。