

平成25年度 修士論文

テストケースに含まれるキーワードに着目した  
テスト自動化スクリプトの作成手法の提案

電気通信大学 大学院

情報理工学研究科 総合情報学専攻

経営情報学コース

学籍番号 1130016

氏名 長田 学

指導教員 西 康晴 講師

副指導教員 鈴木 和幸 教授

提出日 平成 26 年 01 月 30 日

# 目次

<b>1</b>	<b>はじめに.....</b>	<b>1</b>
1.1	ソフトウェアテストの重要性.....	1
1.2	ソフトウェアテストの方法.....	2
1.3	キャプチャアンドプレイバック(C&P)ツールとは.....	5
1.4	キーワード駆動テストの課題.....	7
<b>2</b>	<b>本研究の目的.....</b>	<b>8</b>
<b>3</b>	<b>提案内容.....</b>	<b>9</b>
3.1	手法概要.....	10
3.2	試算.....	12
<b>4</b>	<b>テストスクリプト生成プログラム.....</b>	<b>13</b>
4.1	必要な成果物.....	13
4.2	テストスクリプト生成プログラムの処理の流れ.....	19
<b>5</b>	<b>検証.....</b>	<b>31</b>
5.1	検証内容.....	31
5.2	検証手順.....	33
5.3	テストベースの説明.....	34
5.4	検証結果.....	42
<b>6</b>	<b>考察.....</b>	<b>44</b>
6.1	作成時間の比較.....	44
6.2	作成効率の比較.....	45

6.3	生成プログラムの作成コストの考慮 .....	46
7	まとめ .....	50
8	今後の課題 .....	51
9	参考文献 .....	52
10	付録 .....	53
	謝辞 .....	59

## 1 はじめに

### 1.1 ソフトウェアテストの重要性

ソフトウェア開発においてソフトウェアテストは、製品の品質を確認する大切なアクティビティである。ソフトウェアテストは、時間的・経済的な制約の中で、ソフトウェアに内在する欠陥をより多く見つけ、障害が発生しないことを確認しなければならない。しかし、近年のソフトウェア開発は、ソフトウェアの大規模複雑化や開発サイクルの短縮化により、ソフトウェアテストを効率的に実施することが求められている。

## 1.2 ソフトウェアテストの方法

ソフトウェアテストは、入力に対する処理に対する期待値と実際値を比較することで、その比較対象の処理の実装の妥当性を判断する。ソフトウェアには、ウィンドウやボタンなど UI 要素を持つ GUI アプリケーションと文字ベースの CUI アプリケーションがある。それぞれのアプリケーションにおけるテストについて述べる。

### 1.2.1 CUI アプリケーションのテスト

CUI アプリケーションのテストは、プログラミング言語を用いてテストコードを記述し、プロダクトコードのテストを行う。一般的には、単体テスト・結合テストレベルで開発者が行うテストである。例えば、表 1.1 のような記述を行う。

統合開発環境では、CUI テストをサポートする仕組みが備わっており、容易に CUI テストを作成したり、実行したりできる。

表 1.1 CUI アプリケーションのテストケースの例

```
Class SampleClass
{
    Int WARIZAN(int A, int B)
    {
        if(A ==0 && B==0)
        {
            Throw new ZeroException();
        }
        return A/B;
    }
}

Class SampleClassTest
{
    Void TestWARIZAN()
    {
        this.assert(WARIZAN(10, 5), 2);           //OK
        this.assert(WARIZAN(10, 1), 10);         //OK
        this.assert(WARIZAN(99,0), 0);           //Exception
    }
}
```

## 1.2.2 GUI アプリケーションのテスト

GUI アプリケーションのテストは、CUI アプリケーションのテストに比べ、テスト実施に手間がかかる。例えば、図 1.1 に示すような電卓ソフトをテストする際、 $10 \div 5$  の計算を確かめるためには、ボタンを 5 回押して、表示された結果を人が見て確かめることになるだろう。簡単なテストなら良いが、テストが複雑かつ大量になると、人間が操作ミスを起こす可能性が高まる。

そのため、GUI アプリケーションをテストするためには、人間が動作させたマウスやキーボードなどの入力を記録し、それを再生し、操作を繰り返し行えるキャプチャ&プレイバックツールを用いられる。



図 1.1 GUI アプリケーションの例

### 1.3 キャプチャアンドプレイバック(C&P)ツールとは

GUI アプリケーションのテストは，ユーザーの操作を記録し，その記録した操作をプレイバックするツールを用いる．このツールをキャプチャアンドプレイバック(C&P)ツールと呼ばれる．一般的にC&P ツールで用いられるユーザーの操作情報や入力データ情報は，スクリプト言語を用いて記述されている．

Mark Fewster や Dorothy Graham[1]らは，C&P ツールのスクリプトの再利用性によって，5つのレベルに分類した(表 1.2)．

表 1.2 スクリプトの記述レベル

記述レベル	概要
線形スクリプト	手動のテストをキャプチャして記録する
構造化スクリプト	スクリプトの記述に選択と繰り返しのプログラミング構造が使用できる
共有スクリプト	スクリプトを他のスクリプトから呼び出せる
データ駆動スクリプト	入力データ，検証データなどのデータを外部ファイルに分離したスクリプト
キーワード駆動スクリプト	データに加えて，ユーザーの操作も外部ファイルに分離したスクリプト



### 1.3.1 キーワード駆動テストについて

キーワード駆動スクリプトを使用した GUI テストをキーワード駆動テストと呼ばれる。キーワード駆動テストの概念は、Fewster や Graham[1], Buwalda[2]などによって提案されているが、特に定まった手法があるわけではない。キーワード駆動テストは、ユーザーの操作情報を抽象化したものを”キーワード”と呼び、キーワードを使用して、操作の手順をスプレッドシートに記述することが特徴である。これにより、操作手順やデータをプログラム言語から分離し、プログラミングスキルを持たない人でも、テストスクリプトを理解したり記述したりできるようになるというメリットがある。

図 1.2 は、あるサイトにログインしようとするキーワード駆動テストのテストスクリプトの例である。

	A	B	C	D	E	F
1	Run	Keyword	ObjectDetails	Action	Action2	Comm
2	r	LaunchApp	<a href="https://login.yahoo.co.jp/config/login">https://login.yahoo.co.jp/config/login</a>			
3	r	importdata	<a href="C:\O2T4SWD\config\TestScript\TestData_TestScript1.xls">C:\O2T4SWD\config\TestScript\TestData_TestScript1.xls</a>			
4	r	perform	;Username	set:dt_Username		
5	r	wait		1000		
6	r	perform	;Password	set:dt_Password		
7	r	wait		1000		
8	r	screenshot				
9	r	perform	;Loginbutton	click		
10	r	wait		2000		
11	r	screenshot				
12	r	check	;LabelUsername	linktext:dt_Username		
13						
14						

図 1.2 キーワード駆動テストのテストスクリプトの例

## 1.4 キーワード駆動テストの課題

キーワード駆動テストのテストスクリプトを生成するには、テストケースを参照しながら手動で作成することになる。

既にテストケースを作成し、蓄積しているような組織の場合、とても多くの量のテストケースを保有している。そのような組織で、テストの自動化を行おうとする際に、キーワード駆動テストのテストスクリプトを一つ一つ作成すると、大きな時間と労力がかかることになる。

## 2 本研究の目的

本研究の目的は、ユーザーが自然言語で記述したテストケースからキーワード駆動テストのテストスクリプトを生成し、テストスクリプトの作成するコストを下げることである。

### 3 提案内容

テストケースに記述されているテスト目的に、「キーワード」となる言葉が含まれていることに着目した。例えば、テストケース「ログインして、ユーザーIDの表示を確認する」は、下線の箇所がキーワードやその引数である。

テストケースからキーワードを抽出し、そのキーワードを用いてテストスクリプトの作成をコンピュータで行うことができれば、テストスクリプトの作成にかかるコストを下げることができると考えた。

### 3.1 手法概要

本研究は，テストケースからキーワードを抽出し，そのキーワードを用いてテストスクリプトをコンピュータで作成する手法を提案する．本研究の手法を用いたテストケース生成プログラムを実行するためには，いくつかの入力成果物を事前に作成する必要がある．そのうち，レベル変換表と引数表についても，本手法で提案する．

#### 3.1.1 キーワードについて

テストケースには，テストで留意すべき観点が記述されている．例えば，テストケース「ログインして，ユーザーIDの表示を確認する」は，下線の箇所がテスト観点となる．その中でも，動作を表すテスト観点をキーワードとなる．これらのキーワードは日本語の係り受け解析や形態素解析を用いることで抽出が可能である．

キーワードには，操作レベルとドメインレベルの 2 つのレベルがある．例えば，操作レベルのキーワードは「入力」「押下」などである．また，ドメインレベルのキーワードは，「ログイン」「購入」などである．

操作レベルのキーワードを複数組み合わせ構成される．例えば，「ログイン」は，「ユーザーIDを入力する」「パスワードを入力する」

「ログインボタンを押下する」という操作レベルのキーワードの組み合わせとなる。

## 3.2 試算

100 件のテストケースをキーワード駆動テストのテストスクリプトに変換するケースを考える。

手動でテストスクリプトを作成する場合、テストスクリプトは 100 個必要であり、一つのテストスクリプトあたり 3 分かかるとすれば、約 300 分かかると見積もることができる。

本手法を用いる場合、レベル変換表と引数表の 2 つの成果物を作る必要がある。

どちらの成果物も、全体のテストケースの半分のテストケースを分析すれば作成できると仮定し、一つのテストケースにつきレベル変換表は 3 分、引数表は 1 分かかるとすると、成果物の作成には、約 200 分かかると見積になる。

また、本手法の変換処理はテストケース 1 件 5 秒かかるとして、約 9 分ですべての処理を終える。

手動と本手法を比較すると、約 300 分と約 209 分で、約 3 割の時間を短縮できる試算結果となった。

## 4 テストスクリプト生成プログラム

ユーザーは、事前にいくつかの成果物を用意し、生成プログラムを実行させる。生成プログラムは、テストケースに対するキーワード駆動テストのテストスクリプトを出力する。

今回、出力結果のテストスクリプトは、**Open2Test[3]**という NTT データが管理するキーワード駆動テストのフレームワークを利用する。従って、生成されるキーワード駆動テストのテストスクリプトは **Open2Test** で用いられるテストスクリプトを出力する。

### 4.1 必要な成果物

本手法には、事前に以下の成果物の用意が必要である。

- テストケース
- テストデータ
- オブジェクトマップ
- レベル変換表
- 引数表



### 4.1.1 テストケース

ユーザーが自然言語で記述したテストの要約や目的、テストデータのパス、対象 URI が記述されたドキュメントであり、本手法の入力成果物である。

テスト目的項目に、テストケースがどのようなテストを表すかを一文で記述する。データ項目には、テストケースに対応するテストデータのスプレッドシートのパスを記述する。

	A	C	D	F	G	H
1	ID	テスト目的	備考	期待値	データ	基準URL
2	1	予約申込を行い、宿泊予約確認ページの表示を確認する	デフォルト値を入力する。	入力値と同様の値が表示される	C:\02T4SWD¥conf¥TestScript¥TestData_TestScript1.xls	file:///C:/STARHOTEL/serveApp/index.html
3	2	日付を今日の日付で予約申込を行い、エラーメッセージの表示を確認する	宿泊日以外は、デフォルト値を入力する。	宿泊日には、翌日以降の日付を指定してください。	C:\02T4SWD¥conf¥TestScript¥TestData_TestScript2.xls	file:///C:/STARHOTEL/serveApp/index.html
4	3	日付を不正な日付で予約申込を行い、エラーメッセージの表示を確認する。	不正な日付とは、例えば「2012/13/99」など 宿泊日以外は、デフォルト値を入力する。	宿泊日が間違っています	C:\02T4SWD¥conf¥TestScript¥TestData_TestScript3.xls	file:///C:/STARHOTEL/serveApp/index.html
5	4	年フィールドに任意の文字列を入力し、次へボタンを押下し、エラーメッ	任意の文字列とは、例えば「aaaa」「(スペース)」などである。 宿泊日以外は、デフォルト値	年月日、期間、人数いずれかの値が半角英数の範囲外です	C:\02T4SWD¥conf¥TestScript¥TestData_TestScript4.xls	file:///C:/STARHOTEL/serveApp/index.html

図 4.1 テストケース

### 4.1.2 テストデータ

テストデータは、テストケースのテスト目的に従って、テストケース一件につき、一つのスプレッドシートを作成する。対象となるテストケースの具体的なテストデータが記述される。従来のキーワード駆動テストでも、テストデータは作成する。

先頭行には、列の名前を記述する。

	A	B
1	reserveyear	errorcheckresult
2	aaaa	年月日、期間、人数いずれかの値が半角 英数の範囲外です
3		

図 4.2 テストデータ

### 4.1.3 オブジェクトマップ

オブジェクトマップは、テスト対象の要素を識別する ID とテストスクリプト内で用いられる名称をマッピングする表である。表 4.3 に示す。要素を識別する ID の取得には、開発者の協力を仰ぐ必要がある。従来のキーワード駆動テストでも、テストデータは作成する。

	A	B	C	D	E
1		Object Name	Object Type	Parent	ObjectPath
2		reserveyear			id=reserve_year
3		reservemonth			id=reserve_month
4		reserveday			id=reserve_day
5		reserveterm			id=reserve_term
6		headcount			id=headcount
7		breakfaston			id=breakfast_on
8		breakfastoff			id=breakfast_off
9		plana			id=plan_a
10		planb			id=plan_b
11		guestname			id=guestname
12		gotonext			id=goto_next
13		price			id=price
14		gname			id=gname
15		commit			id=commit
16		succesmsg			xpath=html/body/div[1]/h1
17		errorcheckresult			id=errorcheck_result
18					
19					

図 4.3 オブジェクトマップ

#### 4.1.4 レベル変換表

ドメインレベルのキーワードを操作レベルのキーワードに変換するために使用する。例えば、「ログインする」のレベル変換表は図 4.4 の通りである。レベル変換表は、従来のキーワード駆動テストでは存在しない成果物である。

	A	B	C	D	
1	Domain	Details	Operation	Detail	
2	予約申込		押下	次へボタン	
3	予約申込	日付	入力	年フィールド	
4	予約申込	日付	入力	月フィールド	
5	予約申込	日付	入力	日フィールド	
6	予約申込	日付	押下	次へボタン	
7	確認	料金	押下	次へボタン	
8	確認	料金	確認	料金	
9					
10					

図 4.4 レベル変換表

### 4.1.5 引数表

キーワードには、引数を持つ。その引数を管理するために使用する。一つの引数には、オブジェクトマップに記述された要素の名称とテストデータのスプレッドシートの列名の2つを持つ。(図 4.5)

従来のキーワード駆動テストでは存在しない成果物である。

	A	B	C
1	Details	Path	colname
2	年フィールド	reserveyear	dt_reserveyear
3	次へボタン	gotonext	
4	エラーメッセージ	errorcheckresult	dt_errorcheckresult
5	宿泊予約確認ページ		
6	月フィールド	reservemonth	dt_reservemonth
7	日フィールド	reserveday	dt_reserveday
8	宿泊日数フィールド	reserveterm	dt_reserveterm
9	人数フィールド	headcount	dt_headcount
10	料金	money	dt_money
11	名前フィールド	guestname	dt_guestname
12			

図 4.5 引数表

## 4.2 本手法の流れ

テストスクリプト生成プログラムの処理の流れは、図 4.6 に示す。テストケースが記述されたスプレッドシートを読み込み、読み込むテストケースがなくなるまで生成処理を繰り返す。

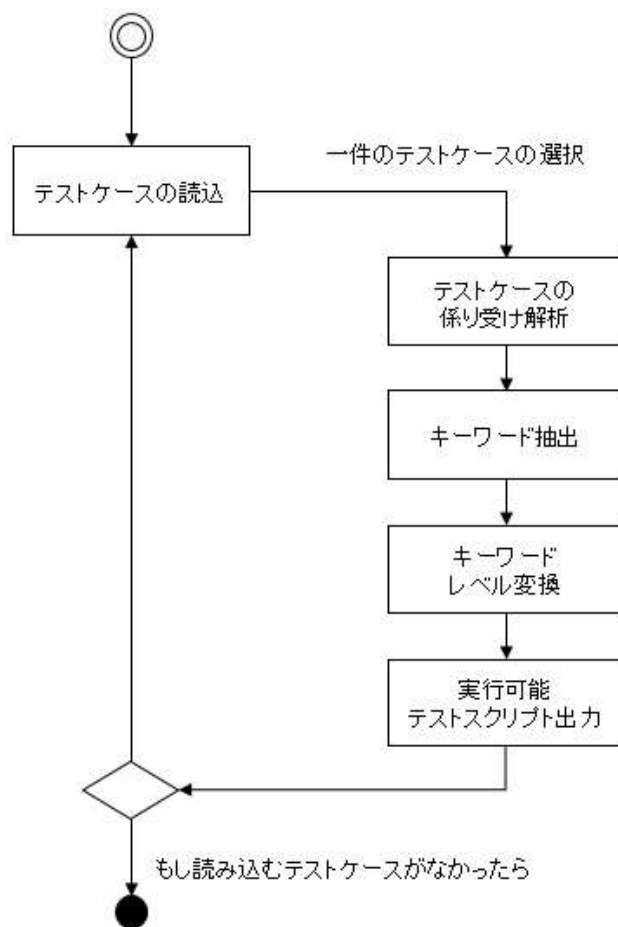


図 4.6 処理の流れ

#### 4.2.1 日本語係り受け解析処理

テストケースのテスト目的に記述された文章を，ヤフー株式会社が提供する WebAPI サービスである「日本語係り受け解析 Web API[4]」を用いて解析を行う。

解析するためには，WebAPI の URL に解析したい文を UTF-8 でエンコーディングし，パラメーターにセットし，リクエストを送る。すると，API は係り受け解析の結果を，XML 形式で送り返してくれる。図 4.7 は，「日付を明日で予約申込を行い，料金を確認する。」をリクエストした結果である。

```
ResponseXML.xml
<?xml version="1.0" encoding="UTF-8"?>
<ResultSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:yahoo:jp:j">
  <Result>
    <ChunkList>
      <Chunk>
        <Id>0</Id>
        <Dependency>3</Dependency>
        <MorphemList>
          <Morphem>
            <Surface>日付</Surface>
            <Reading>ひづけ</Reading>
            <Baseform>日付</Baseform>
            <POS>名詞</POS>
            <Feature>名詞,名詞,*,日付,ひづけ,日付</Feature>
          </Morphem>
          <Morphem>
            <Surface>を</Surface>
            <Reading>を</Reading>
            <Baseform>を</Baseform>
            <POS>助詞</POS>
            <Feature>助詞,格助詞,*,を,を,を</Feature>
          </Morphem>
        </MorphemList>
      </Chunk>
      <Chunk>
        <Id>1</Id>
        <Dependency>3</Dependency>
        <MorphemList>
          <Morphem>
            <Surface>明日</Surface>
            <Reading>あした</Reading>
            <Baseform>明日</Baseform>
            <POS>名詞</POS>
            <Feature>名詞,名詞,*,明日,あした,明日</Feature>
          </Morphem>
          <Morphem>
            <Surface>で</Surface>
            <Reading>で</Reading>
            <Baseform>で</Baseform>
            <POS>助詞</POS>
            <Feature>助詞,格助詞,*,で,で,で</Feature>
          </Morphem>
        </MorphemList>
      </Chunk>
      <Chunk>
        <Id>2</Id>
        <Dependency>3</Dependency>
        <MorphemList>
          <Morphem>
            <Surface>予約</Surface>
            <Reading>よやく</Reading>
            <Baseform>予約</Baseform>
            <POS>名詞</POS>
          </Morphem>
        </MorphemList>
      </Chunk>
    </ChunkList>
  </Result>
</ResultSet>
```

図 4.7 WebAPI のリクエスト結果



## 4.2.2 キーワード抽出

日本語係り受け解析処理の XML を用いて、キーワードとそれに紐付く情報の抽出を行う。

キーワード候補は、必ず係り受けの係先になる。そのため、「予約申込を行い、料金を確認する。」の例では、「行い,」「確認する。」の文節がキーワード候補となる。キーワード候補の文節に動詞、助動詞が含まれる(表 4.1)。

次に、キーワードを判断する。キーワード候補の文節に含まれる動詞、助動詞がどのような単語かを判断し、その単語によって、キーワードの判定の処理を変える。

表 4.2 に判定ルールを示す.

キーワードが判定されたら, キーワードと同様にキーワードに紐付く情報を抽出する. このキーワードに紐付く情報のことを引数と呼ぶ. 引数の取得ルールを表 4.3 に示す. キーワード「入力」と「確認」だけ特別なルールがある.

キーワードの抽出処理が終わると, 「日付を明日で予約申込を行い, 料金を確認する.」の例では, 表 4.4 に示すようなデータが生成される.

表 4.1 係り受け解析の結果

文節	係り先の文節	キーワード候補
日付を	-> 行い,	
明日で	-> 行い,	
予約申込を	-> 行い,	
行い,	-> 確認する.	○
料金を	-> 確認する.	
確認する.	-> 文末	○

表 4.2 キーワードの判断ルール

基本形	品詞	処理	キーワード例
行う	動詞	動詞「行う」に係る文節 の中で助詞「を」を持つ 文節の名詞句を抽出する	「予約申込を行う」 ⇒「予約申込」
する	助動詞	「する」「し」の前にある 名詞句を抽出する	「～を確認する。」 ⇒「確認」 「入力し、～」 ⇒「入力」

表 4.3 引数の取得ルール

キーワード	ルール	例
(任意)	キーワード候補の文節にかかる文節で、助詞「を」を持つ文節を選択し、その名詞句抽出する	「日付を明日の日付で予約申込をし、～」 ⇒「日付」
入力	キーワード候補の文節にかかる文節で、助詞「に」を持つ文節を選択し、その名詞句抽出する	「名前フィールドに ABC と入力し、～」 ⇒「名前フィールド」
確認	キーワード候補の文節にかかる文節で、助詞「を」を持つ文節を選択し、その名詞句抽出する。その際、抽出された引数が、名詞「表示」の場合、「表示」の文節にかかる文節の中から、助詞「の」を持つ文節の名詞句を抽出する	「エラーメッセージの表示を確認する」 ⇒「エラーメッセージ」

表 4.4 中間テストスクリプトの例

キーワード	引数
予約申込	日付
確認	料金

### 4.2.3 キーワードレベル変換処理

ここでは，ドメインレベルのキーワードを操作レベルのキーワードに変換する．ユーザーは，事前に表 4.5 のようなドメインレベルと操作レベルのキーワードの関係を表すレベル変換表を作成する．

中間テストスクリプトのキーワードと引数の組み合わせが，レベル変換表のドメインレベルキーワードと引数に存在する場合，該当する中間テストスクリプトのキーワードと引数を，操作レベルキーワードと引数に置換する(表 4.6)．

表 4.5 レベル変換表の例

ドメインレベル キーワード	引数	操作レベル キーワード	引数
予約申込		押下	次へボタン
予約申込	人数	入力	人数フィールド
予約申込	人数	押下	次へボタン
予約申込	日付	入力	年フィールド
予約申込	日付	入力	月フィールド
予約申込	日付	入力	日フィールド
予約申込	日付	押下	次へボタン

表 4.6 レベル変換後の中間テストスクリプトの例

キーワード	引数
入力	年
入力	月
入力	日
押下	次へボタン
確認	料金

#### 4.2.4 実行可能なテストスクリプトへの変換処理

ここでは、中間テストスクリプトから実際にキーワード駆動テストで実行可能なテストスクリプトに変換する。ユーザーは、事前に表 4.7 のような引数に紐付く情報を記述した引数表を作成する。各引数には、オブジェクト名とデータ列名を持つ。オブジェクト名は、オブジェクトマップに記述したオブジェクト名前を記述する。また、データ列名は、テストデータの列の名称を記述する。ボタンなどの「押下」キーワードに紐付く引数には、データ列名は記述されない。

実行可能なテストスクリプトへの変換は、キーワードごとに事前に開発者が変換処理をプログラムで実装しておく必要がある。この変換処理と引数表を用いて、実行可能なテストスクリプトを生成する。実行可能なテストスクリプトの先頭には、テスト先の URL、テストデータの格納されたファイルのパス、初期値などが既に入力されている。出力結果は図 4.8 のようになる。



表 4.7 引数表の例

引数名	オブジェクト名	データ列名
年フィールド	reserveyear	dt_eserveyear
月フィールド	reserveyear	dt_reserveyear
日フィールド	reserveyear	dt_eserveyear
次へボタン	gotonext	
人数フィールド	headcount	dt_headcount
料金	price	dt_price

	A	B	C	D	E	F	G
1	run	keyword	details	action1			
2	r	launchApp	file:///C:/STARHOTEL/reserveApp/index.html				
3	r	importdata	C:%02T4\$WD#config#TestScript#TestData_TestScript1.xls				
4	r	perform	;reserveyear	set:2014			
5	r	perform	;reservemonth	set:1			
6	r	perform	;reserveday	set:23			
7	r	perform	;reserveterm	set:1			
8	r	perform	;headcount	set:1			
9	r	perform	;guestname	set:osada			
10	r	perform	;breakfastoff	Check:0n			
11	r	perform	;reserveyear	set:dt_reserveyear			
12	r	perform	;reservemonth	set:dt_reservemonth			
13	r	perform	;reserveday	set:dt_reserveday			
14	r	perform	;gotonext	click			
15	r	wait		1000			
16	r	check	;money	linktext:dt_money			
17							

図 4.8 実行可能なテストスクリプトの例

## 5 検証

検証では、手動でテストケースからテストスクリプトを作成する場合と本手法を用いたプログラムを用いてテストスクリプトを作成する場合において、テストスクリプトを作成するために必要なユーザーの作業時間を測定し、比較した。本手法では、手法に用いるテストスクリプト生成プログラムは、事前に作成してある前提である。

### 5.1 検証内容

被験者には、あらかじめ用意したテストベースに対してテストケースを記述してもらい、そのテストケースから、手動でキーワード駆動テストのテストスクリプトの作成する場合とテストスクリプト生成プログラムを用いる作成する場合の2つを行ってもらう。

被験者には、ソフトウェア開発を行ったことのない被験者 N とソフトウェア開発経験 1 年目の被験者 S の 2 人を用意した。

測定項目として、2つの手法のテストスクリプトを生成するまでにかかった時間を測定する。

テストスクリプト生成プログラムを用いる方法では、開発者によって生成プログラムは既に作られている前提である。今回の検証ではこのプログラムの作成にかかった時間は含めず、ユーザーがプロ

グラムを動かすために必要な成果物を作成する作業時間を測定する.

## 5.2 検証手順

被験者に対して、以下の手順で作業を依頼した。手順2と手順4において、作業時間の測定を行う。

### 手順1：テストケースの作成

テストベースとなるホテル予約システムの説明を行い、このシステムに対するテストケースを記述するように依頼をする。

### 手順2：テストスクリプトの作成

キーワード駆動テストの説明を行い、作成したテストケースからキーワード駆動テストのテストスクリプトを作成するように依頼する。このテストスクリプトの作成にかかった作業時間を測定する。

### 手順3：テストケースの修正

生成プログラムのキーワードの抽出方法についての説明を行い、キーワード抽出が困難なテストケースは書き直すように依頼する。

### 手順4：レベル変換表，引数表の作成

生成プログラムを動作させるために必要なレベル変換表と引数表を作成する。この二つの成果物を作成にかかった作業時間を測定する。

### 5.3 テストベースの説明

テストベースには，テスト自動化研究会のサイト[4]で公開されているホテルの予約システムを用いる．

予約システムは，3つのページから構成する(表 5.1)．

予約申込ページから，予約確認ページへ遷移する処理を，「予約申込」とし，予約確認ページから予約確定ページへ遷移する処理を「予約確定」とする(図 5.1)．

表 5.1 ページ一覧

ページ名	ファイル名
予約フォームページ	index.html
予約確認ページ	check_info.html
予約確定ページ	final_confirm.html

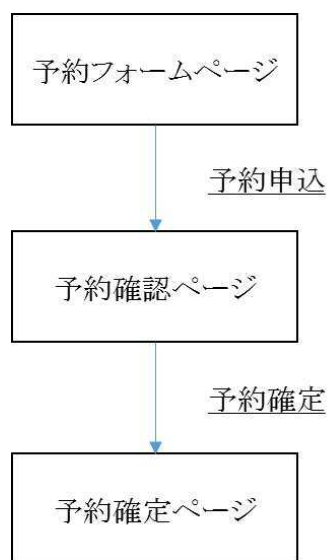


図 5.1 ページ遷移

### 5.3.1 予約フォームページ

予約フォームページは、予約の情報を入力するページである。入力する情報は、宿泊年、宿泊月、宿泊日、宿泊日数、宿泊人数、名前の6項目である。これらの情報を入力後、次へボタンを押下すると、予約確認ページへ遷移する

検証に用いたホテルの予約システムは、テストケースの量を調整するために一部ソースコードを変更した。変更した行は、`index.html`の24行目から31行にあるHTMLタグのスタイル属性に対して、`"visibility:hidden"`の値を指定した。また、HTMLから外れている文字列は削除した。図5.2に予約フォームページを示す。

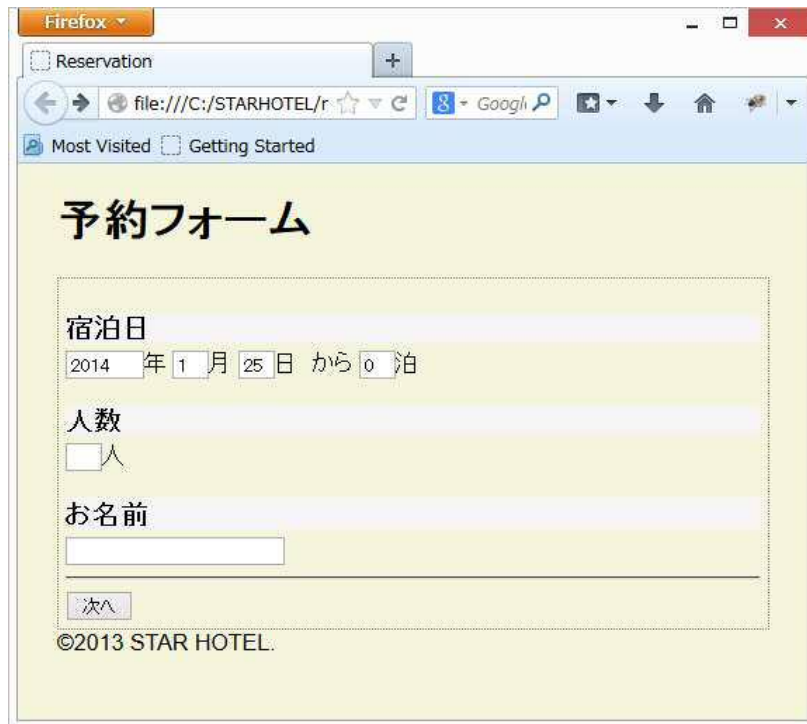


図 5.2 予約フォームページ



### 5.3.2 予約確認ページ

予約フォームページで入力した情報を検証し，問題なければ入力された情報を表示し，ユーザーが予約情報の確認を行えるようにする(図 5.3). 予約情報の確認後，確定ボタンを押下すると，予約確定ページに遷移する．もし，入力された情報に問題がある場合，エラーメッセージを表示する(図 5.4).

入力情報の検証内容は，表 5.2 に示す.



図 5.3 予約確認ページ

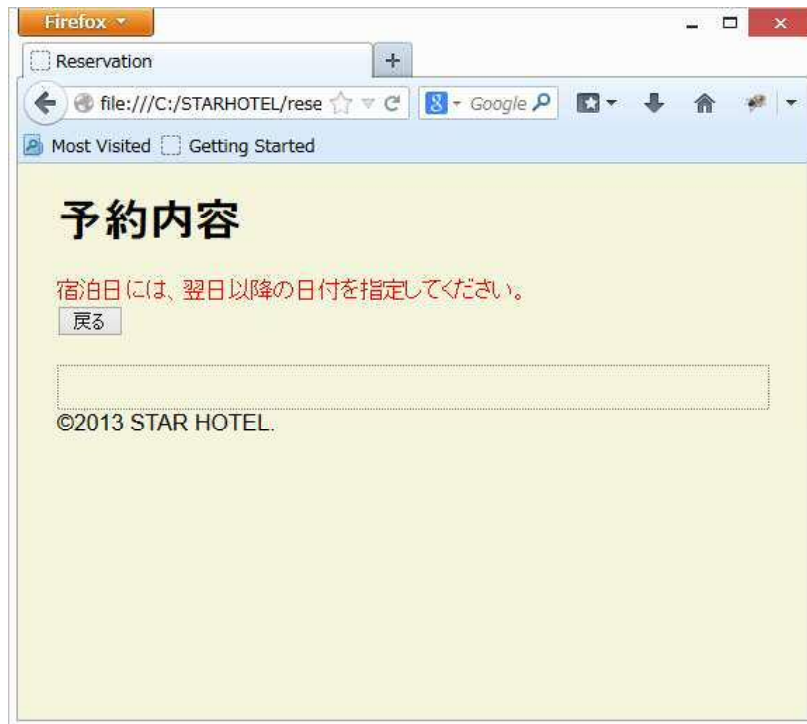


図 5.4 エラーメッセージの表示

表 5.2 バリデーション一覧

エラーメッセージ	検査条件
宿泊日には、翌日以降の日付を指定してください.	日付が今日以前の場合(今日を含む)
宿泊日が間違っています.	2013/13/99 など日付としておかしい場合
宿泊日には、3ヶ月以内のお日にちのみ指定できます.	日付が今日から3ヶ月以降の場合
年月日、期間、人数いずれかの値が半角英数の範囲外です.	年フィールド、月フィールド、日フィールド、宿泊日数フィールド、人数フィールドのどれかに半角英数字以外の文字を入力した場合
年月日、期間、人数、またはお名前のいずれかが空です.	年フィールド、月フィールド、日フィールド、宿泊日数フィールド、人数フィールドのどれかのフィールドが未入力の場合
宿泊日数が1日以下です.	宿泊日数フィールドに0を入力した場合
人数が入力されていません.	宿泊人数フィールドに0を入力した場合

### 5.3.3 予約確定ページ

予約確定ページは、予約完了のメッセージを表示する(図 5.5)

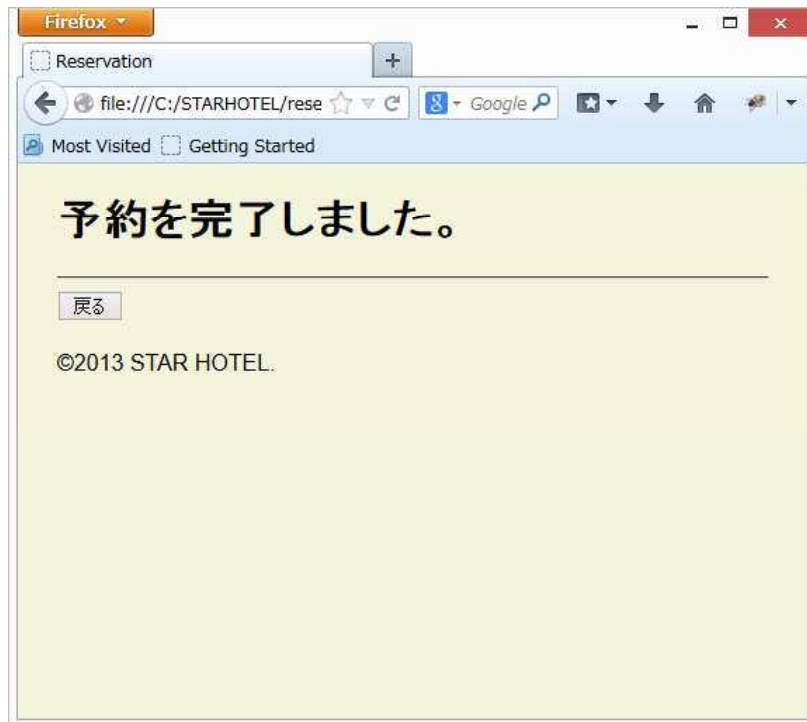


図 5.5 予約確定ページ

## 5.4 検証結果

### 5.4.1 テストケース数

最初に被験者にテストベースを対象にして，テストケースを作成してもらった．表 5.3 はそのテストケース数を示す．

表 5.3 テストケース数

被験者	テストケースの数
N	36
S	27

### 5.4.2 手動でテストスクリプトを作成する場合

作成してもらったテストケース 1 件に対して，キーワード駆動テストで実行可能なテストスクリプトを手動で作成してもらった．その作成にかかった時間を表 5.4 に示す．

表 5.4 手動作成にかかった時間

被験者	作成時間
N	82 分 22 秒
S	67 分 10 秒

### 5.4.3 生成プログラムを用いて作成する場合

生成プログラムの入力情報として必要なレベル変換表と引数表を作成してもらい，その作成にかかった時間を測定した．また，生成プログラムの実行時間も記録した(表 5.5)

表 5.5 生成プログラムを用いた場合にかかった時間

被験者	レベル変換表の 作成時間	引数表の 作成時間	プログラムの 実行時間	合計
N	4分47秒	5分17秒	58秒	11分32秒
S	9分18秒	16分42秒	42秒	26分42秒

## 6 考察

### 6.1 作成時間の比較

表 6.1 に、検証での手動で作成した場合と生成プログラムを用いて作成した場合の作業時間をまとめた。これにより、適切にキーワードが含まれたテストケースが記述されている場合、どの被験者もテストケースの作成は手動で作成するより、生成プログラムを用いて作成した方が短い時間で作成できることを確認できた。

表 6.1 手法にかかった時間の比較

被験者	テスト ケース数	手動で作成	生成プログラムを 用いて作成	短縮率
N	36	82 分 22 秒	11 分 32 秒	87%
S	27	67 分 10 秒	26 分 42 秒	60%

## 6.2 作成効率の比較

作成効率は、テストケースの作成時間をテストケース数で除算したものと定義し、手法ごとの作成効率を表 6.2 にまとめた。

表 6.2 より、手動で作成した場合の作成効率は、被験者によって大きな差は現れなかったが、生成プログラムを用いて作成した場合、被験者 N と被験者 S の間には、約 3 倍の差が現れることがわかった。

表 6.2 テストスクリプトの作成効率

被験者	手動で作成	生成プログラムを用いて作成
N	137sec	18sec
S	149sec	59sec

これは、被験者のテストケースに出現するビジネスレベルのキーワードの多さに関係すると考えられる。総テストケースの中でビジネスレベルのキーワードの出現回を数えると、被験者 N は 1 件に対して、被験者 S は 27 件と大きく差があることがわかる。



### 6.3 生成プログラムの作成コストの考慮

今回の検証では、生成プログラムは予め作成された状態で行い、生成プログラムの作成のコストは考慮していない。そこで、生成プログラムの作成コストを加味して、手動で作る場合と生成プログラムで作る場合のどちらがよいのかを考察する。

#### 6.3.1 生成プログラムの作成コストの試算

検証プログラムの作成コストを試算するために、生成プログラムを機能分解し、その設計・実装にかかる時間の見積もりをした。表 6.3 に見積もり結果を示す。表 6.3 の結果より、生成プログラムの設計・実装には、約 14 時間かかる見積もりになった。

この生成プログラムの作成見積と被験者 S のテストスクリプトの作成効率を用いて、損益分岐点を算出した(数式 6.1)。

#### 数式 6.1 損益分岐点の算出

$$\begin{cases} y = 149x \\ y = 59x + 825 * 60 \end{cases}$$
$$149x = 59x + 49500$$
$$x = \frac{49500}{149 - 59} = 825$$

算出した結果から、テストケース数が 825 に満たない場合は、手動でテストスクリプトを作成した方が工数は小さく、テストケース

数が 825 以上の場合は，本手法を用いてテストケースを生成した方が工数は小さいことがわかった．

表 6.3 生成プログラムの見積

タスク	詳細	時間[min]
全体設計	プログラムの仕様を決める	60
詳細設計	テストケース取込処理	60
詳細設計	キーワード抽出処理	120
詳細設計	レベル変換処理	60
詳細設計	スクリプト生成処理	60
実装	テストケース取込処理>エクセル読込	15
テスト	テストケース取込処理	15
実装	キーワード抽出処理> 係り受け解析のリクエスト	15
実装	キーワード抽出処理>解析結果の分析	240
実装	キーワード抽出処理>中間データの作成	15
テスト	キーワード抽出処理	15
実装	レベル変換処理>エクセル読込	15
実装	レベル変換処理>レベル変換	30
テスト	レベル変換処理	15
実装	スクリプト生成処理>エクセル読込	15
実装	スクリプト生成処理>スクリプト生成	30
実装	スクリプト生成処理>エクセル書き出し	15
テスト	スクリプト生成処理	15
テスト	全体を通して動作させる	15
	合計	825

### 6.3.2 本手法の有用性

IPA が公表しているソフトウェア開発データ白書 2012-2013[6]には、工数当たりのテストケース数の分析結果がある。それによると、新規開発の場合の工数当たりの結合テストケース数の中央値は、123.6[件/1000 人時]である。この数字を参考にして、開発にかかる工数の規模の損益分岐点を計算した(数式 6.2)。

#### 数式 6.2 開発にかかる工数の規模の損益分岐点

$$\frac{825[\text{件}]}{123.6[\text{件}/1000 \text{ 人時}]} = 6675[\text{人時}]$$

数式 6.2 より、開発にかかる工数が 6675 人時を下回る場合、手動でテストスクリプトを作成した方がよく、6675 人時を上回る時は本手法を用いる方がよいという結果になった。

また、ソフトウェア開発データ白書 2012-2013 には、ソフトウェア開発のプロジェクトの工数の調査結果がある。それによると、調査したプロジェクト(N=3089)における基本設計からベンダ確認のシステムテストまでの工程の実績値の中央値は、7893[人時]となっている。従って、本手法は 2012-2013 年に実施されたプロジェクトの大半に対して有用性を持つことがわかる。

## 7 まとめ

本研究では，自然言語で記述されたテストケースから，キーワード駆動テストで実行可能なテストスクリプトを生成する手法を提案した．

検証では，本研究の手法を用いる場合と用いない場合のキーワード駆動テストのキーワードを作成する作業時間の測定を行い，本手法を用いた場合，ユーザーの作業時間を短縮できることを確認した．

考察では，テストケース生成プログラムの開発時間も含めて本手法を有用性について考察し，本手法は 825 件以上のテストケースを扱う場合に有用であることがわかった．

## 8 今後の課題

テストケースの記述の抽象度が高い場合，キーワードに変換は行っても，テストスクリプトとして成り立たないことがある．

例えば，検証で被験者 S が最初に書いたテストスクリプトから抜粋すると，被験者 S は「人数が 0 で予約できないことを確認する」という記述をした．現状の生成プログラムでキーワード抽出をすると，下線の箇所のみがキーワードとして抽出され，実行可能なキーワード駆動テストのテストスクリプトを生成できない．

組織がこのように抽象度が高いテストケースを保有している場合，本手法を用いるためには，2つの方法がある．一つは，本手法で使用する生成プログラムのキーワードの抽出ルールを保有しているテストケースに合わせて開発する方法と，もう一つはテストケースを書き直す方法である．どちらの方法を選択するかは，テストケースの量やキーワードの抽出ルールの難易度を考慮して決めるべきである．

## 9 参考文献

- [1] Mark Fewster; Dorothy Graham. "Software Test Automation". 1st ed, ACM Press, 1999, 600p.
- [2] Hans Buwalda. "Action Based Testing". stickyminds.com. <http://www.stickyminds.com/better-software-magazine-article/not-just-number-real-value-metrics>, (accessed 2014-01-29)
- [3] Open2Test.org." Welcome to Open2Test.org". Open2Test. <http://www.open2test.org/index.html>, (accessed 2014-01-29)
- [4] ヤフー株式会社," テキスト解析:日本語係り受け解析 API - Yahoo!デベロッパーネットワーク".Yahoo! デベロッパーネットワーク . <http://developer.yahoo.co.jp/webapi/jlp/da/v1/parse.html>, (accessed 2014-01-29)
- [5] テスト自動化研究会 , " SoftwareTestAutomationResearch(STAR) GitHub " .github.<https://github.com/SoftwareTestAutomationResearch>, (accessed 2014-01-29)
- [6] IPA 独立行政法人 情報処理推進機構," SEC BOOKS : ソフトウェア開発データ白書 2012-2013 : IPA 独立行政法人 情報処理推進機構 ".IPA . <http://www.ipa.go.jp/sec/publish/tn12-002.html>, (accessed 2014-01-29)

## 10 付録

検証で作成した被験者ごとにテストケースの要約を掲載する。

**表 10.1 被験者 N の手順 2 の時のテストケース**

ID	テスト目的
1	規定された日程以外の日付が入力されたときの表示を確認する
2	一桁の月のときに、二桁で入力したときの表示を確認する
3	お名前に環境依存文字を入力したときの表示を確認する
4	宿泊者名に記号を入力する
5	宿泊数に 0 を入力したときの表示を確認する
6	宿泊数に全角の数字で入力したときの表示を確認する
7	宿泊数に全角の記号を入力したときの表示を確認する
8	宿泊数に全角の文字を入力したときの表示を確認する
9	宿泊数に半角の記号を入力したときの表示を確認する
10	宿泊数に半角の文字を入力したときの表示を確認する
11	宿泊月に 0 を入力したときの表示を確認する
12	宿泊月に記号と数字の組み合わせを入力したときの表示を確認する
13	宿泊月に全角で入力したときの表示を確認する
14	宿泊月に全角の記号を入力したときの表示を確認する
15	宿泊月に全角の文字を入力したときの表示を確認する
16	宿泊月に半角の記号を入力したときの表示を確認する
17	宿泊人数に 0 を入力したときの表示を確認する
18	宿泊人数に全角の記号を入力したときの表示を確認する
19	宿泊人数に全角の文字を入力したときの表示を確認する
20	宿泊人数に半角の記号を入力したときの表示を確認する
21	宿泊人数に半角の文字を入力したときの表示を確認する
22	宿泊年に 0 を入力したときの表示を確認する
23	宿泊年に記号と数字の組み合わせを入力したときの表示を確認する
24	宿泊年に全角で入力したときの表示を確認する
25	宿泊年に全角の記号を入力したときの表示を確認する



---

26	宿泊年に全角の文字を入力したときの表示を確認する
27	宿泊年に半角の記号を入力したときの表示を確認する
28	宿泊日に0を入力したときの表示を確認する
29	宿泊日に記号と数字の組み合わせを入力したときの表示を確認する
30	宿泊日に全角で入力したときの表示を確認する
31	宿泊日に全角の記号を入力したときの表示を確認する
32	宿泊日に全角の文字を入力したときの表示を確認する
33	宿泊日に半角の記号を入力したときの表示を確認する
34	宿泊日に半角の文字を入力したときの表示を確認する
35	存在しない月を入力したときの表示を確認する
36	存在しない日を入力したときの表示を確認する
37	予約申し込みができ、予約確定を押したときの表示を確認する

---

表 10.2 被験者 N の手順 4 の時のテストケース

ID	キーワード
1	宿泊日に規定された日程以外の日付を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
2	宿泊月に月の前に0をつけて入力し、次へボタンを押下し、エラーメッセージの表示を確認する
3	宿泊者名に環境依存文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
4	宿泊者名に記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
5	宿泊数に0を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
6	宿泊数に全角で入力し、次へボタンを押下し、エラーメッセージの表示を確認する
7	宿泊数に全角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
8	宿泊数に全角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
9	宿泊数に半角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
10	宿泊数に半角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
11	宿泊月に0を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
12	宿泊月に記号と数字の組み合わせを入力し、次へボタンを押下し、エラーメッセージの表示を確認する
13	宿泊月に全角で入力し、次へボタンを押下し、エラーメッセージの表示を確認する
14	宿泊月に全角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
15	宿泊月に全角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
16	宿泊月に半角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
17	宿泊人数に0を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
18	宿泊人数に全角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
19	宿泊人数に全角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
20	宿泊人数に半角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
21	宿泊人数に半角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
22	宿泊年に0を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
23	宿泊年に記号と数字の組み合わせを入力し、次へボタンを押下し、エラーメッセージの表示を確認する
24	宿泊年に全角で入力し、次へボタンを押下し、エラーメッセージの表示を確認する
25	宿泊年に全角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
26	宿泊年に全角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
27	宿泊年に半角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
28	宿泊日に0を入力し、次へボタンを押下し、エラーメッセージの表示を確認する

- 
- 29 宿泊日に記号と数字の組み合わせを入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 30 宿泊日に全角で入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 31 宿泊日に全角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 32 宿泊日に全角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 33 宿泊日に半角の記号を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 34 宿泊日に半角の文字を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 35 宿泊月に存在しない月を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 36 宿泊日に存在しない日を入力し、次へボタンを押下し、エラーメッセージの表示を確認する
- 
- 37 予約申込をし、予約確定をし、確定メッセージの表示を確認する
-

**表 10.3 被験者 S の手順 2 の時のテストケース**

ID	テスト目的
1	予約申込が正常に動作する。
2	予約確定が正常に動作する
3	月が 13 以上で予約できないことを確認する
4	月が未入力で予約できないことを確認する
5	月に文字を入力して予約できないことを確認する
6	3 ヶ月以降の予約ができない
7	3 ヶ月後の予約ができる
8	名前が全角 50 文字まで入力できることを確認する
9	名前が全角半角 50 文字まで入力できることを確認する
10	名前が半角 50 文字まで入力できることを確認する
11	名前が未入力で予約できないことを確認する
12	人数が 0 で予約できないことを確認する
13	人数が 9 で予約できることを確認する
14	人が未入力で予約できないことを確認する
15	人に文字を入力して予約できないことを確認する
16	年だけ来年で予約できないことを確認する
17	年が未入力で予約できないことを確認する
18	年に文字を入力して予約できないことを確認する
19	当日の予約ができない
20	過去の日付で予約できない
21	泊数が 0 で予約できないことを確認する
22	泊数が 9 で予約できることを確認する
23	泊が未入力で予約できないことを確認する
24	泊に文字を入力して予約できないことを確認する
25	日が 32 以上で予約できないことを確認する
26	日が未入力で予約できないことを確認する
27	日に文字を入力して予約できないことを確認する

表 10.4 被験者 S の手順 4 の時のテストケース

ID	テスト目的
1	予約申込を行い、ページの表示を確認する
2	予約確定を行い、確定メッセージを確認する
3	月を 13 で予約申込を行い、エラーメッセージを確認する
4	月を空文字で予約申込を行い、エラーメッセージを確認する
5	月を aa で予約申込を行い、エラーメッセージを確認する
6	日付を 3 ヶ月後以降で予約申込を行い、エラーメッセージを確認する
7	日付を 3 ヶ月前で予約申込を行い、エラーメッセージを確認する
8	名前を全角 50 文字で予約申込を行い、ページの表示を確認する
9	名前を全角半角混じりの 50 文字で予約申込を行い、ページの表示を確認する
10	名前を半角 50 文字で予約申込を行い、ページの表示を確認する
11	名前を空文字で予約申込を行い、エラーメッセージを確認する
12	人数を 0 で予約申込を行い、エラーメッセージを確認する
13	人数を 9 で予約申込を行い、ページの表示を確認する
14	人数を空行で予約申込を行い、エラーメッセージを確認する
15	人数を a で予約申込を行い、エラーメッセージを確認する
16	日付を来年の今日で予約申込を行い、エラーメッセージを確認する
17	年を空行で予約申込を行い、エラーメッセージを確認する
18	年を aaaa で予約申込を行い、エラーメッセージを確認する
19	日付を今日で予約申込を行い、エラーメッセージを確認する
20	日付を前日で予約申込を行い、エラーメッセージを確認する
21	泊数を 0 で予約申込を行い、エラーメッセージを確認する
22	泊数を 9 で予約申込を行い、ページの表示を確認する
23	泊数を空行で予約申込を行い、エラーメッセージを確認する
24	泊数を a で予約申込を行い、エラーメッセージを確認する
25	日を 32 で予約申込を行い、エラーメッセージを確認する
26	日を空行で予約申込を行い、エラーメッセージを確認する
27	日を aa で予約申込を行い、エラーメッセージを確認する

## 謝辞

本論文の執筆にあたり，優しく指導して下さった西康晴先生には心から感謝致します．ものを直接的には教えてくれませんが，振る舞いから伝わる考え方やマインドはとても勉強になります．いつか先生のような考え方ができるように精進します．

また，研究活動に理解を示し，職場で色々と融通して下さった斉藤仁さんにも心から感謝致します．斉藤さんが上司でなければ，職業と学業を掛け持つことは難しかったです．修士論文終わったら，仕事も程よくがんばります．

大学院入学から3年間様々なことがありましたが，研究室を離れることなく無事ここまで辿り着けたのは，にし研究室や職場の皆さんの多くの支援があったからこそだと思っています．お陰様で無事に論文を書き上げることができました．

皆さんには，本当に感謝致します．ありがとうございました．