

Metamodeling Construction for Generating Test Case via Decision Table Based on Korean Requirement Specifications

Woo Sung Jang[†] · So Young Moon^{††} · R. Young Chul Kim^{†††}

ABSTRACT

Many existing test case generation researchers extract test cases from models. However, research on generating test cases from natural language requirements is required in practice. For this purpose, the combination of natural language analysis and requirements engineering is very necessary. However, Requirements analysis written in Korean is difficult due to the diverse meaning of sentence expressions. We research test case generation through natural language requirement definition analysis, C3Tree model, cause-effect graph, and decision table steps as one of the test case generation methods from Korean natural requirements. As an intermediate step, this paper generates test cases from C3Tree model-based decision tables using meta-modeling. This method has the advantage of being able to easily maintain the model-to-model and model-to-text transformation processes by modifying only the transformation rules. If an existing model is modified or a new model is added, only the model transformation rules can be maintained without changing the program algorithm. As a result of the evaluation, all combinations for the decision table were automatically generated as test cases.

Keywords : Meta Modeling, Korean Requirement Analysis, Test Case Automatic Generation, Cause-Effect Graph, Decision Table, C3Tree(Conditional and Conjunction Clause Tree)

한글 요구사항 기반 결정 테이블로부터 테스트 케이스 생성을 위한 메타모델링 구축화

장우성[†] · 문소영^{††} · 김영철^{†††}

요약

기존의 다양한 테스트 케이스 생성에 대한 연구는 모델로부터 테스트 케이스를 추출한다. 하지만 실무의 경우 자연어 요구사항 문장으로부터 테스트 케이스를 생성할 수 있어야 한다. 이를 위해 자연어 문장의 분석하고, 분석 과정 및 결과를 요구공학 영역에 접목하는 연구는 매우 필요하다. 하지만 한국어 문장의 다양성 때문에, 한국어 자연어 요구사항 분석은 어려운 이슈이다. 우리는 한국어 자연어 요구사항으로부터 테스트 케이스 생성 연구 중 하나로서, 자연어 요구사항의 정의 분석, C3Tree 모델의 생성, 원인-결과 그래프의 생성, 결정 테이블의 생성 단계를 통한 테스트 케이스 생성 방법을 연구한다. 본 논문은 중단 단계로써, 메타모델링 변환 기법을 이용하여 C3Tree 모델 기반의 결정 테이블로부터 테스트 케이스 생성 방법을 제안한다. 이 방법은 모델 변환 규칙의 수정을 통해 모델 to 모델, 모델 to 텍스트로의 변환 과정을 제어한다. 모델이 변경되거나, 새로운 모델이 추가되더라도 프로그램 알고리즘의 직접적인 수정 없이 모델 변환 규칙을 유지보수 할 수 있다. 평가 결과, 결정 테이블에 대한 모든 조합이 테스트 케이스로 자동 생성되었다.

키워드 : 메타모델링, 한국어 요구사항 분석, 테스트 케이스 자동 생성, 원인-결과 그래프, 결정 테이블, C3Tree(Conditional and Conjunction Clause Tree)

※ 본 연구는 2023년도 문화체육관광부의 재원으로 한국콘텐츠진흥원(과제명: 인공지능 기반 사용자 대화형 멀티모달 인터랙티브 스토리텔링 3D장면 저작 기술 개발, 과제번호: RS-2023-00227917, 기여율:50%) 지원과 2023년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업(과제명: NLP BE RT Model 기반 자동 리팩토링을 통한 무결점 코드화 연구, 과제번호: No. 2021R111A3050407, 기여율: 40%)의 지원과 2022년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업(과제명: 비정형 요구사항 명세서 기반 자동 비용 예측 및 역공학을 통한 검증 연구, 과제번호: No.2021R111A1A01044060, 기여율:10%)의 지원을 받아 수행된 연구임.

※ 이 논문은 2023년 소프트웨어공학연구회의 우수논문으로 "자연어 요구사항 기반 테스트케이스 생성을 위한 중간 모델들의 메타모델링 방법"의 제목으로 발표된 논문을 확장한 것임.

† 준 회 원 : 홍익대학교 소프트웨어융합학과 박사후과정

†† 비 회 원 : 홍익대학교 소프트웨어융합학과 조빙교수

††† 총신회원 : 홍익대학교 소프트웨어융합학과 교수

Manuscript Received : May 4, 2023

First Revision : July 7, 2023

Accepted : July 24, 2023

* Corresponding Author : R. Young Chul Kim(bob@hongik.ac.kr)

1. 서론

테스트 케이스의 자동 생성은 테스트의 비용을 크게 낮출 수 있는 요소 중 하나이다. 영어 기반 요구사항 분석은 명사(일반동사, 대명사), 동사(be-동사, have-동사, 일반 동사)의 추출이 쉽다. 한국 언어학의 경우, 품사, 조사의 종류 및 표현, 특히 어간, 어미 등의 다양성으로 명료하게 필요한 속성들을 추출하기 어려운 점이 있다. 이러한 이유 때문에 자연어 요구사항의 분석이 어렵다. 기존 한국어 요구사항 분석 연구는 정형화된 요구사항 문장을 분석한다[1]. 그리고 인공지능 기반 요구사항 분석 연구는 기초적인 수준으로 비교/분석이 진행되고 있다[2]. 하지만 한국어 요구사항에서 원인/결과 속성을 자동 추출하고, 원인-결과 그래프 생성하여 테스트 케이스를

자동 생성하는 연구는 찾기 어렵다.

이러한 문제의 해결을 위해, 우리는 한국어 자연어 요구사항으로부터 테스트 케이스 자동 생성 방법을 연구한다. 한국어 자연어 요구사항으로부터 Conditional and Conjunction Clause Tree(C3Tree) 모델들을 생성하고, 여러 C3Tree 모델들을 하나로 통합하여 C3Tree 모델로부터 원인-결과 그래프를 생성한다[3]. 원인-결과 그래프로부터 결정 테이블을 생성하고[4], 결정 테이블로부터 테스트케이스를 생성한다. 이 방법은 최소한의 테스트케이스로 100% 커버리지를 만족할 수 있다[5,6]. 이 방법은 인수테스트에 대한 비용을 낮출 수 있다.

이 논문은 연구 수행의 중간 산출물로서, C3Tree 모델을 기반으로 생성된 결정 테이블로부터 테스트 케이스 자동 생성 방법[7] 및 구축 환경을 제안한다. 제안하는 방법은 1) 결정 테이블 모델과 테스트 케이스 모델을 위한 메타모델[8]을 생성한다. 2) 생성된 메타모델들 간의 변환 규칙을 설계한다. 3) 변환 규칙을 실행하여 메타모델 간의 변환을 수행하는 메타모델 변환 엔진을 설계한다. 4) 메타모델 변환 엔진을 실행하여 결정 테이블 모델을 테스트 케이스 모델로 자동 변환한다. 이 방법은 모델이 수정 또는 추가된다면, 프로그램의 알고리즘을 수정 없이 메타모델 파일과 변환 규칙 파일의 추가만을 통해 프로그램을 유지보수 할 수 있는 장점을 가진다[9].

본 논문의 구성은 다음과 같다. 2장은 관련 연구로서 요구사항으로부터 테스트 케이스 자동 생성 프로세스와 메타모델 변환 엔진을 언급한다. 3장은 C3Tree 모델을 기반으로 생성된 결정 테이블로부터 테스트 케이스를 자동 생성하는 방법을 설계한다. 4장은 C3Tree 모델을 기반으로 생성된 결정 테이블로부터 테스트 케이스를 자동 생성하는 방법을 구현한다. 5장은 실험 결과를 언급한다. 6장은 결론 및 향후 연구를 언급한다.

2. 관련 연구

2.1 한국어 자연어 요구사항으로부터 테스트 케이스 생성 및 실행 프로세스

Fig. 1은 한국어 자연어 요구사항으로부터 테스트 케이스 생성 및 실행 프로세스를 보여준다[3,4].

Step1은 한국어 요구사항을 입력받고, 요구사항 문장의 형태소를 분석한다.

Step2는 분석된 형태소를 통해 복잡한 문장을 간소화한다. 문장에서 조건절, 접속절을 식별하여 문장을 절 단위로 나눈다. 이때 절 식별 우선순위 알고리즘을 통해 조건절, 접속절 중 먼저 나뉘어야 될 절을 식별한다. 나뉜 절은 작은 문장으로 변환되어 말뭉치 정규화가 적용된다. 생략된 주어가 복구되고, 수동적인 문장은 능동적인 문장으로 변환된다.

Step3은 Step2에서 문장이 간소화되는 과정을 C3Tree 모델로 표현한다. C3Tree 모델은 복잡한 문장이 단순한 형

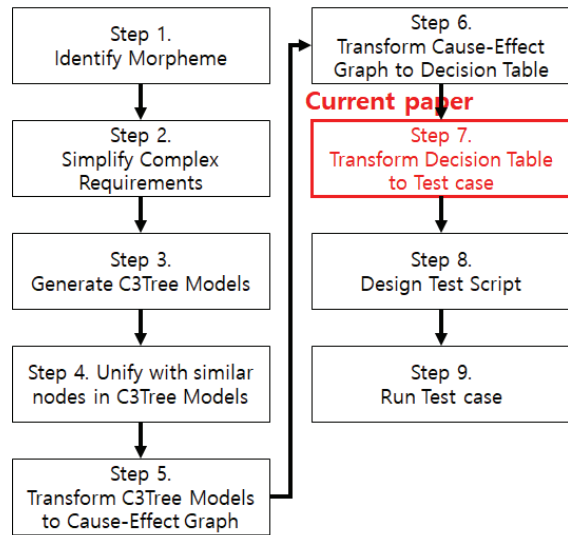


Fig. 1. Automatically Test Case Generation and Execution Process from Korean Natural Requirements

태의 문장으로 나뉘는 과정을 이진 트리 형태로 표현한다. 결과적으로 요구사항 문장 하나 당 하나의 C3Tree 모델이 생성된다.

Step4는 여러 개의 C3Tree 모델을 하나의 C3Tree 모델로 통합한다. 자카드 유사도, 의미역 분석 기법을 통해 C3Tree 모델 내 노드들의 유사성을 식별하여 유사한 노드들은 하나의 노드로 합친다. 결과적으로 하나의 통합된 C3Tree 모델을 생성한다.

Step5는 통합된 C3Tree 모델을 기반으로 원인-결과 그래프를 생성한다.

Step 6는 메타모델 변환 기법을 통해 원인-결과 그래프를 결정 테이블로 변환한다.

Step 7은 메타모델 변환 기법을 통해 결정 테이블을 테스트 케이스로 변환한다.

Step 8은 테스트 케이스의 테스트 스크립트를 생성한다.

Step 9는 테스트 스크립트를 실시간 실행하여 실제 타겟 프로그램을 동적 테스트한다.

2.2 한국어 요구사항 기반 테스트 케이스 생성 방법

Fig. 2는 요구사항 정형화 및 테스트 케이스와 테스트 스크립트 생성 방안이다[1]. 이 연구는 STE 철학 기반의 정형화된 한국어 요구사항 작성을 지원한다. 또한 연결어미의 제한이 존재한다.

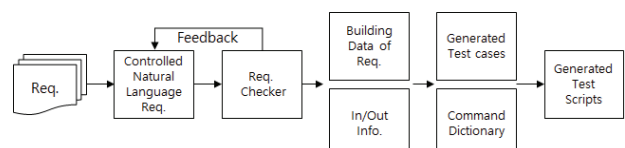


Fig. 2. Test case Generation Process[1]

3. C3Tree 모델 기반의 결정 테이블로부터 테스트 케이스 자동 생성 방법 설계

3.1 결정 테이블 to 테스트 케이스 모델 변환 메커니즘

Fig. 3 하단의 Model Translator는 결정 테이블 모델로부터 테스트 케이스 모델을 자동 생성하는 모델 변환 메커니즘을 가진다. 모델 변환엔진은 결정 테이블 메타모델을 준수하는 결정 테이블 모델 읽고, 변환 규칙을 실행하여, 테스트 케이스 메타모델을 준수하는 테스트 케이스 모델을 생성한다.

Fig. 3 상단의 Decision Table Metamodel은 결정 테이블 메타모델의 구조이다. DecisionTable은 여러 개의 Rule과 Element를 포함한다.

Element 클래스는 결정 테이블 내 항목들의 정보를 저장한다. ceid는 원인-결과 그래프 노드들의 고유번호이다. ceid가 동일한 Element들이 있다면, 이 Element들은 precondition 관계를 가진다. type이 input인 Element가 실행되기 위해서는 type이 output인 Element의 입력 조합을 true로 출력되도록 수행해야 한다. deid는 결정 테이블 내 Input 또는 Output 항목의 고유번호이다. type은 Element의 속성으로써 input 또는 output의 값을 가진다. value는 자연어 문장이다.

Rule 클래스는 Input과 Output의 조합 정보를 가진다.

Input 클래스는 결정 테이블의 Input 항목들의 정보를 저장한다. deid는 타겟이 된 Element의 고유번호를 저장한다. value는 해당 Input 항목의 값의 true인지 false인지 저장한다. True는 Y가 저장된다. False는 N이 저장된다.

Output 클래스는 결정 테이블의 Output 항목들의 정보를 저장한다. 속성에 대한 설정은 Input과 같다.

Fig. 4는 결정 테이블 메타모델로부터 결정 테이블 생성 예시이다. 5개의 Element를 생성하고, 각 Element의 조합의 일부를 보여준다. 메타모델은 XMI Code로 생성될 수 있다. XMI 코드는 뷰에 의해 파싱되어 결정 테이블로 보여질 수 있다.

Fig. 3 중단의 Testcase Metamodel은 테스트 케이스 메타모델의 구조이다. Testcase는 여러 개의 Case 클래스 정보를 포함한다.

Case 클래스는 테스트케이스의 여러 개의 Precondition, Send, Recv를 포함한다. drid 속성은 결정 테이블 내 Element의 고유번호이다.

Precondition 클래스는 테스트케이스의 실행을 위한 사전 조건이다. 결정 테이블에서 미리 실행되어야 하는 조합을 식별하여 생성된다. value는 자연어 문장이다.

Send 클래스는 테스트 프로그램이 테스트케이스의 입력 데이터이다. 각 속성은 Precondition과 같다.

Recv 클래스는 테스트케이스의 예상 결과이다. 각 속성은 Precondition과 같다.

3.2 변환 규칙 정의

Fig. 5는 결정 테이블 메타모델과 테스트 케이스 메타모델 간의 변환 과정이다. Rule은 Case로 변환된다. Precondition

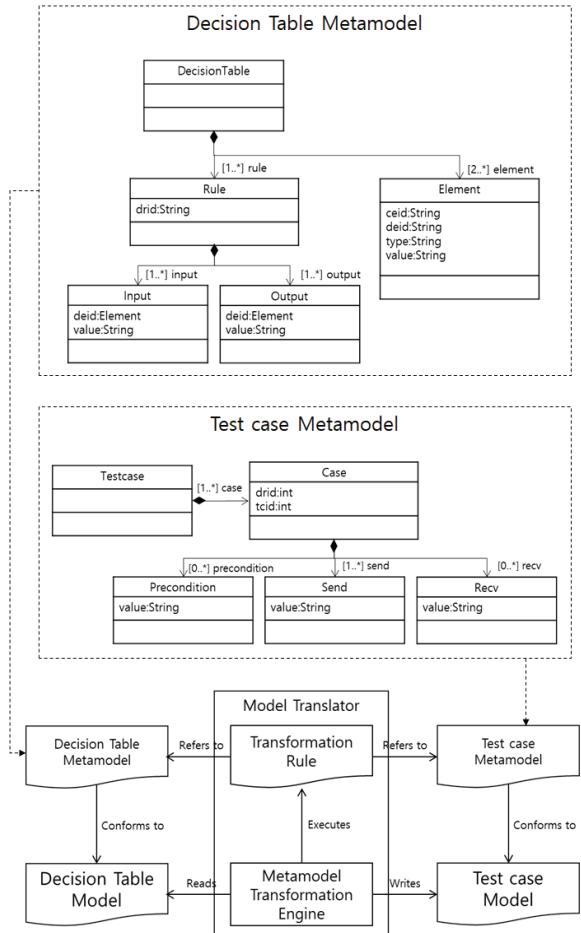
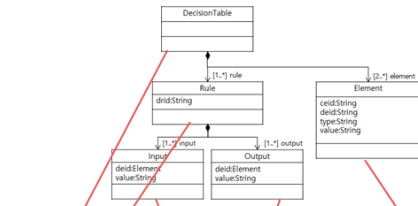


Fig. 3. Decision Table Model to Test Case Model Transformation Mechanism

Decision Table Meta model



Decision Table Model XMI Code

```
<?xml version="1.0" encoding="UTF-8"?>
<DecisionTable>
  <element ceid="0" deid="0" type="input" value="The user enters a number"/>
  <element ceid="1" deid="1" type="input" value="The user presses the enter key"/>
  <element ceid="2" deid="2" type="input" value="The program pops up an alert window"/>
  <element ceid="3" deid="3" type="output" value="The program pops up an alert window"/>
  <element ceid="4" deid="4" type="output" value="An alarm sounds"/>
  <rule drid="0">
    <input deid="0" value="y"/>
    <input deid="1" value="y"/>
    <input deid="3" value="n"/>
    <output deid="2" value="y"/>
    <output deid="4" value="n"/>
  </rule>
</DecisionTable>
```

Decision Table

Input	Rule0	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7
The user enters a number	Y	N	Y	Y	Y	N	N	N
The user presses the enter key	Y	N	Y	N	N	Y	Y	N
The program pops up an alert window	N	Y	Y	Y	N	Y	N	N
Output								
The program pops up an alert window	Y	N	N	N	N	N	N	N
An alarm sounds	N	N	Y	N	N	N	N	N

Fig. 4. Decision Table Generation Process from Decision Table Metamodel

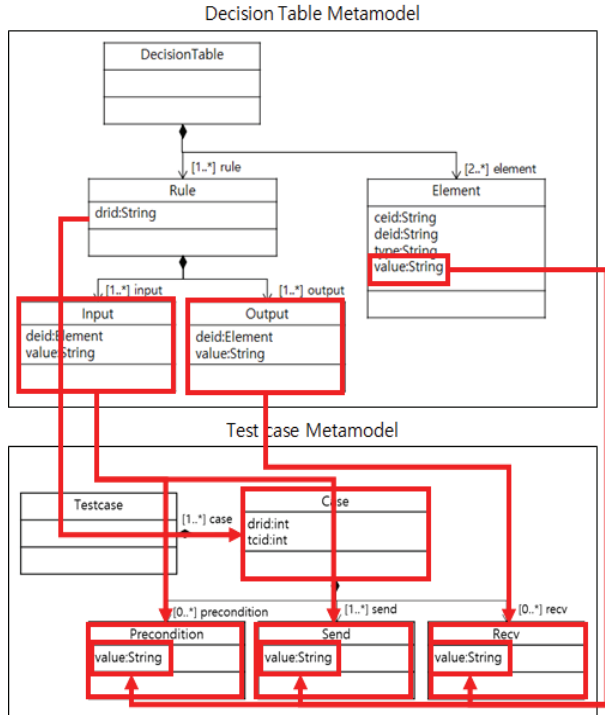


Fig. 5. Model transformation rule

은 결정 테이블에서 동일한 의미를 가지는 Input과 Output이 있을 때, Input의 값이 true인 경우, 해당 항목의 Output을 true로 만드는 Rule의 Input 조합으로 만들어진다.

3.3 모델의 XML 코드

Table 1은 결정 테이블 모델의 XML 코드 예제이다. DecisionTable 태그 내에 n개의 element 태그, rule 태그가 포함된다. rule 태그 내에 n개의 input 태그, output 태그가 포함된다. 각 태그는 메타모델에서 정의된 속성 정보를 포함한다.

Table 2는 테스트 케이스 모델의 XML 코드 예제이다. Testcase 태그 내에 n개의 case 태그가 포함된다. case 태그 내에 n개의 send태그, rcv 태그, precondition 태그가 포함된다. 각 태그는 메타모델에서 정의된 속성 정보를 포함한다.

Table 1. Decision table XML code example

```
<?xml version="1.0" encoding="UTF-8"?>
<DecisionTable>
  <element ceid="0" deid="0" type="input" value="Sentence1"/>
  <element ceid="1" deid="1" type="input" value="Sentence2"/>
  ...
  <rule drid="0">
    <input deid="0" value="y"/>
    <input deid="1" value="y"/>
    <input deid="3" value="n"/>
    <output deid="2" value="y"/>
    <output deid="4" value="n"/>
  </rule>
  ...
</DecisionTable>
```

Table 2. Test Case XML Code Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Testcase>
  <case drid="0" tcid="0">
    <send value="Sentence1"/>
    <send value="Sentence2"/>
    <rcv value="Sentence3"/>
  </case>
  <case drid="1" tcid="1">
    <send value="Sentence1"/>
    <rcv value="Sentence2"/>
    <precondition value="Sentence3"/>
    ...
  </case>
  ...
</Testcase>
```

Decision Table

Input	Rule0	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7
The user enters a number	Y	N	Y	Y	Y	N	N	N
The user presses the enter key	Y	N	Y	N	N	Y	Y	N
The program pops up an alert window	N	Y	Y	Y	N	Y	N	N
Output								
The program pops up an alert window	Y	N	N	N	N	N	N	N
an alarm sounds	N	Y	N	N	N	N	N	N

Test case

Testcase ID	Precondition	Input	Expected Output
0		The user enters a number The user presses the enter key	The program pops up an alert window
1		The user enters a number The user presses the enter key	an alarm sounds
2		The user enters a number The user presses the enter key	
3		The user enters a number The user presses the enter key	
4		The user enters a number	
5		The user enters a number The user presses the enter key	
6		The user presses the enter key	

Fig. 6. Transformation Example from Decision Table to Test Case

Fig. 6은 결정 테이블로부터 테스트 케이스 생성 예시이다. 결정 테이블의 조합이 테스트 케이스로 생성된다. Rule7은 모든 항목이 False이므로 테스트 케이스로 생성되지 않는다.

4. C3Tree 모델 기반의 결정 테이블로부터 테스트 케이스 자동 생성 방법 구현

Fig. 7은 한국어 요구사항 분석기의 실행 환경이다. 한국어 요구사항 분석기의 실행 환경은 라이브러리와 두 개의 모듈 (Korean Analyzer, Model Translator)로 구성된다.

라이브러리 구성요소에서 Mecab-ko는 한국어 문장 형태소를 분석한다. Exobrain은 한국어 의미를 분석한다. Apache와 PHP는 프로그램 기능을 웹서비스 형태로 제공한다. Chart.js

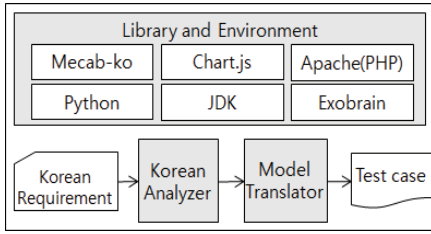


Fig. 7. Korean Requirement Analyzer Environment

는 XMI 코드를 그래프로 변환한다. Python은 한국어 분석 라이브러리를 실행한다. JDK는 Korean Analyzer, Model Translator를 실행한다.

Korean Analyzer는 입력된 한국어 문장으로부터 C3Tree 모델을 생성하고, 여러개의 C3Tree 모델을 하나로 통합하여 원인-결과 그래프를 생성한다.

Model Translator는 원인-결과 그래프를 결정 테이블로 변환하고, 결정 테이블을 테스트 케이스로 변환한다.

5. 실험 결과

기본적인 요구사항 문장을 이용한 실험 결과는 다음과 같다. 모든 요구사항 문장이 작은 절로 나뉘었고, 각 절의 모든 조합이 결정 테이블로 자동 변환되었다. 그리고 결정 테이블의 Input 조합이 테스트 케이스의 Input으로 모두 변환되었고, 결정 테이블의 Output이 테스트 케이스의 Expected Output으로 모두 변환되었다.

요구사항 문장으로부터 C3Tree 모델이 정확하게 생성된다면, 원인-결과 그래프, 결정 테이블, 테스트 케이스가 100% 정확도로 모델 변환되었다.

Table 3은 두 개의 한국어 자연어 요구사항 문장 리스트이다. Table 4는 요구사항 문장으로부터 자동 생성된 결정 테이블 모델이다. Table 5는 결정 테이블로부터 자동 생성된 테스트 케이스이다. 결정테이블 내 Input의 “입력된 내용을 저장한다”=Y인 Rule들의 경우, Output의 “입력된 내용을 저장한다”=Y인 Rule0을 참조하여 precondition으로 생성한다.

Table 3. Korean Requirements

Language	Sentence
Korean	사용자가 텍스트를 입력하고 Save 버튼을 누르면 입력된 내용을 저장한다. 입력된 내용이 저장되면 프로그램을 종료한다.
English	If the user enters text and clicks the Save button, the entered text is saved. If the entered text are saved, close the program.

Table 4. Decision Table Model

List	Rule0	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7
Input								
사용자가 텍스트를 입력하다.	Y	N	Y	Y	Y	N	N	N
사용자가 Save 버튼을 누르다.	Y	N	Y	N	N	Y	Y	N
입력된 내용을 저장한다.	N	Y	Y	Y	N	Y	N	N
Output								
입력된 내용을 저장한다.	Y	N	N	N	N	N	N	N
프로그램이 종료된다.	N	Y	N	N	N	N	N	N

Table 5. Test Case Model

Testcase ID	Precondition	Input	Expected Output
0		사용자가 텍스트를 입력하다. 사용자가 save 버튼을 누르다.	입력된 내용을 저장한다.
1	사용자가 텍스트를 입력하다. 사용자가 save 버튼을 누르다.	입력된 내용을 저장한다.	프로그램이 종료된다.
2	사용자가 텍스트를 입력하다. 사용자가 save 버튼을 누르다.	사용자가 텍스트를 입력하다. 사용자가 save 버튼을 누르다. 입력된 내용을 저장한다.	
3	사용자가 텍스트를 입력하다. 사용자가 save 버튼을 누르다.	사용자가 텍스트를 입력하다. 입력된 내용을 저장한다.	
4		사용자가 텍스트를 입력하다.	
5	사용자가 텍스트를 입력하다. 사용자가 save 버튼을 누르다.	사용자가 save 버튼을 누르다. 입력된 내용을 저장한다.	
6		사용자가 save 버튼을 누르다.	

6. 결론 및 향후 연구

본 논문은 한국어 자연어 요구사항으로부터 테스트 케이스를 자동 생성하기 위한 과정 중 C3Tree 모델 기반 결정 테이블로부터 테스트 케이스를 자동 생성하는 과정을 제안한다. 이 방법은 MDA 기반의 메타모델 변환 방법을 통해 결정 테이블 모델을 테스트 케이스 모델로 자동 변환한다. 중간 모델이 추가된다면, 모델 변환 규칙 파일과 메타모델 파일만을 추가하여 쉽게 모델 변환 과정을 유지보수 할 수 있다. 요구사항 문장으로부터 C3Tree 모델이 정확하게 생성된다면, 원인-결과 그래프, 결정 테이블, 테스트 케이스가 100% 정확하게 모델 변환할 수 있다.

향후 연구로써, 테스트 실행 자동화 연구를 수행하고, 물기반의 테스트 실행 프로세스를 인공지능 기반의 테스트 실행 프로세스로 개선하여 테스트 케이스 생성 정확도를 향상 시킬 예정이다.

References

[1] H. J. Han, K. H. Chung, and K. H. Choi, "Generating test cases and scripts from requirements in controlled language," *KIPS Transactions on Software and Data Engineering*, Vol.8, No.8, pp.331-342, 2019.

[2] B. S. Cho and S. W. Lee, "A comparative study on requirements analysis techniques using natural language processing and machine learning," *Journal of the Korea Society of Computer and Information*, Vol.25, No.7, pp.27-37, 2020.

[3] W. S. Jang and R. Y. C. Kim, "Automatic generation mechanism of cause-effect graph with informal requirement specification based on the Korean language," *Applied Sciences*, Vol.11, Iss.24, 2021.

[4] W. S. Jang and R. Y. C. Kim, "Metamodeling approach for decision table generation from cause-effect graph at the intermediate stages of automatic test case generation," *Korean Institute of Smart Media 2022 Conference*, pp.26-28, 2022.

[5] G. L. Myers, "The Art of Software Testing," John Wiley & Sons, Inc., New Jersey, United States of America, 1979.

[6] G. E. Mogyorodi, "Requirements-Based Testing-Cause-Effect Graphing," In *Software Testing Services: Ontario*, Canada, 2005.

[7] W. S. Jang, Y. S. Kim, H. D. Heo, S. H. Chun, and R. Y. C. Kim, "Metamodeling approach on intermediate models for test case generation from with informal requirement specifications," *Korea Conference on Software Engineering (KCSE) 2023*, Vol.25, No.1, pp.107-108, 2023.

[8] MDA MOF [Internet], <https://www.omg.org/spec/MOF>.

[9] MDA Guide rev. 2.0 [Internet], <https://www.omg.org/cgi-bin/doc?ormsc/14-06-01>.



장우성

<https://orcid.org/0000-0002-8964-1327>

e-mail : uriel200@hongik.ac.kr

2022년 홍익대학교 전자전산공학과

(석·박사)

2022년~현 재 홍익대학교

소프트웨어융합학과 박사후과정

관심분야 : 소프트웨어공학, 임베디드 소프트웨어, 소프트웨어 테스트, 클라우드 시스템, 한국어 분석



문소영

<https://orcid.org/0009-0000-9498-9689>

e-mail : whit2@hongik.ac.kr

2007년~2012년 (주)엑트 대리

2018년 홍익대학교 전자전산공학과

(석·박사)

2017년~현 재 NIPA SP 연구원

2019년~현 재 홍익대학교 소프트웨어융합학과 초빙교수

관심분야 : Software Visualization, Requirement Engineering



김영철

<https://orcid.org/0000-0002-2147-5713>

e-mail : bob@hongik.ac.kr

2000년 Illinois Institute of

Technology(IIT)(공학박사)

2000년~2001년 LG산전중앙연구소

Embedded system 부장

2001년~현 재 홍익대학교 소프트웨어융합학과 교수

관심분야 : 테스트 성숙도 모델(TMM), 모델 기반 테스트,

소프트웨어 가시화(소프트웨어 프로세스, 문서화, 코드

가시화), 저전력, 성능, 정적분석기 개발, 신재생에너지

통합관리 시스템