

## | Oral Session II : Smart Media, Contents/Services

좌장 : 신창선 (순천대)

**Rotation Awareness for Scene Text Detection**

My-Tham Dinh, Guee-Sang Lee (전남대학교) ..... 18

**밀키트용 다회형 기물 용기의 탈부착형 손잡이 개발 방향**

윤봉식(남부대학교), 박수진((주)이씨티) ..... 22

**멀티모달 기반 AI 자동차 분류기에 관한 연구**

이상현, 오승민, 박정훈 (아주대학교) ..... 24

**자동 테스트케이스 중간단계 Cause-Effect Graph로부터 Decision Table 생성을 위한 Metamodeling 기법**

장우성, 김영철 (홍익대학교) ..... 26

**NFT, AI가 융합된 클라이언트 간 상호작용이 가능한 3D 기반 메타버스 서비스**

장은철, 김영철 (홍익대학교) ..... 29

**AI 인물 특징을 활용한 캐릭터 생성에 관한 연구**

여현정, 유원, 문희정(호남대학교) ..... 31

## | Oral Session IX : Big Data, Smart Energy ICT, Smart Information

좌장 : 신춘성 (전남대)

**업종별 부채 예측 모델 개발 : 코로나 19 상황에서**

김양석, 노미진, 김차미, 손승연, 조유진 (계명대학교) ..... 114

**녹조 발생 예측 AI모델 개발 연구**

송수영, 송유선, 이유진, 홍경석, 김남호, 최광미 (호남대학교), 정희자(휴넷가이아) ..... 116

**Non-IID 환경에서 연합 학습 기반 전기 수요 예측**

염성웅, Kolekar Shivani Sanjay, 조현준, 김경백 (전남대학교) ..... 118

**유사 서비스 함수를 위한 코드 모듈들의 구조 내 저전력 연구**

윤예동, 문소영, 김영철 (홍익대학교) ..... 120

**복잡한 코드의 간결화를 통한 성능 및 저전력 개선**

조재형, 문소영, 김영철 (홍익대학교) ..... 123

**CCTV 영상처리를 통한 화재감지기 오탐 개선에 관한 연구**

황은호, 김남호 (호남대학교) ..... 126



**Knowledge Graph 확장을 위한 딥러닝 기반 관계 추출**

최준호, 김형주 (조선대학교) ..... 209

**농경지 침수 분석을 위한 SWMM 모형의 적용성 검토**

김규민, 원다윗, 양승원 (우석대학교) ..... 211

**공간정보 기반 농경지 침수피해의 선제적 대응을 위한 기초자료 구축**

박석우, 양승원, 나인호 (군산대학교) ..... 213

**SWMM 해석 기반 공간분석 농경지 침수의 선제적 대응 연구**

손성민, 김형진 (전북대학교) ..... 215

**색 추출 기법을 접목한 아트 플랫폼의 기대효과**

유세빈, 황시준, 박남홍 (조선대학교) ..... 217

**알츠하이머병에 라지 스케일 네트워크의 연결 패턴 분석**

라마라매쉬쿠마, 권구락 (조선대학교) ..... 219

**클라우드 컴퓨팅에서의 장애 허용 기법 분석**

조만규, 이재환, 김찬수, 박상오 (중앙대학교) ..... 222

**기능점수 기반 정교한 비용 예측 추출을 위한 요구사항 스펙 구조화**

문소영, 김영철 (홍익대학교) ..... 224

**신재생에너지 스마트팜 환경 기반 에너지 사용량 예측**

임종현, 장경민, 오한별, 이명배, 신창선, 박장우, 조용윤 (순천대학교) ..... 226

**Firebase 클라우드 메시징을 활용한 스마트 헬스케어 플랫폼**

남재경, 최민 (충북대학교), 김성준(중원대학교) ..... 228

**수경재배 양액관리를 위한 스마트 단말 모니터링 및 제어 시스템 구현**

오한별, 이명배, 박장우, 조용윤, 신창선 (순천대학교) ..... 230

**데이터 분석 기반의 파프리카 온실 환경 예측에 대한 연구**

장경민, 이명배, 조용윤, 신창선, 박장우 (순천대학교) ..... 232

**딥러닝 모델을 이용한 발전량 예측 방법**

김지인, 이건우, 권구락 (조선대학교) ..... 234

**AMI 시스템에서 수집 시간 단축을 위한 기법 연구**

나채훈, 김정인, 윤범식, 강향숙, 김판구 (조선대학교) ..... 236

# 자동 테스트케이스 중간 단계 Cause-Effect Graph로부터 Decision Table 생성을 위한 Metamodeling 기법

장우성<sup>1\*</sup>, 김영철<sup>2</sup>

홍익대학교 소프트웨어융합학과 소프트웨어공학 연구실

e-mail : {uriel200<sup>1</sup>, bob<sup>2</sup>}@hongik.ac.kr

## Metamodeling Approach for Decision Table Generation from Cause-Effect Graph at the Intermediate Stages of Automatic Test Case Generation

Woo Sung Jang, R. Young Chul Kim

SE Lab, Dept. of Software and Communications Engineering, Hongik University

### 요 약

소프트웨어 요구공학 측면에서, 비정형 요구사항의 명료화 및 테스트케이스 자동 추출은 매우 중요한 이슈이다. 이를 위해 비정형화 요구사항 단계로 부터 Cause-Effect Graph, Decision Table 단계를 거쳐 테스트 케이스를 생성하는 메커니즘 연구는 존재하지만, 자동화를 제공하지 않는다. 또한 요구사항의 자동 추적성을 제공하지 않는다. 문제의 해결을 위해 한국어 요구사항으로부터 Cause-Effect Graph 자동 생성을 통한 Decision Table로 자동 변환하는 방법을 제안한다. 변환 과정에 MDA의 메타모델 기반 변환 방법을 적용한다. 결과적으로 요구사항으로부터 테스트케이스 자동 생성 과정을 위한 추적성을 제공한다.

Keywords: MDA, Meta-Modeling, Cause-Effect Graph, Decision Table, Automatic Test Case Generation

### 1. 서 론

소프트웨어 테스트는 지속적으로 중요성이 증가되고 있다. 하지만 중소기업의 경우 부족한 개발 시간과 테스트 기술 등의 이유로 테스트가 어렵다[1]. 요구사항으로부터 테스트케이스 자동 생성은 기존 문제를 해결할 수 있다. 하지만 한국어 분석의 어려움 때문에 자동화가 어렵다.

우리는 기존 어려움의 해결을 위해 한국어 요구사항으로부터 테스트케이스 자동 생성 방법 연구한다. 1) 한국어 요구사항으로부터 Cause-Effect Graph 자동 생성, 2) Cause-Effect Graph로부터 Decision Table 자동 생성, 3) Decision Table로부터 테스트 케이스를 자동 생성한다. 이 방법은 최소한의 테스트 케이스로 100% 테스트 커버리지를 만족한다[2]. 우리는 기존에 한국어 요구사항 기반 Cause-Effect Graph 자동 생성 방법을 연구하였다[3].

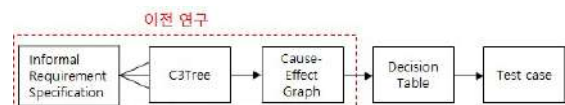
많은 기존 연구들이 Cause-Effect Graph로부터 Decision Table 자동 생성 방법을 제안하지만[2,4,5], 한국어 요구사항에 대한 추적성 적용은 고려하지 않는다. 본 논문은 한국어 요구사항 추적이 가능한 Cause-Effect Graph로부터 Decision Table 자동 생성 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장은 관련 연구를 언급한다. 3장은 메타모델 기반의 Cause-Effect Graph로부터 Decision Table 자동 발생 과정을 언급한다. 4장은 실험

결과를 언급한다. 마지막으로 결론을 언급한다.

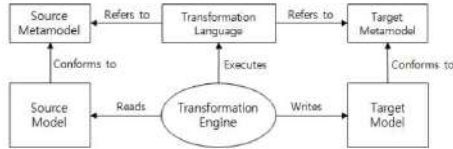
### 2. 관련 연구

한국어 요구사항으로부터 테스트 케이스 자동 발생 방법[3]은 그림 1과 같다. 비정형 요구사항 문장을 간소화한다. 간소화 과정을 C3Tree Model[3]로 표현한다. C3Tree Model에서 원인과 결과의 조합을 찾아 Cause-Effect Graph를 생성한다. Cause-Effect Graph로부터 Decision Table을 거쳐 테스트 케이스를 생성한다.



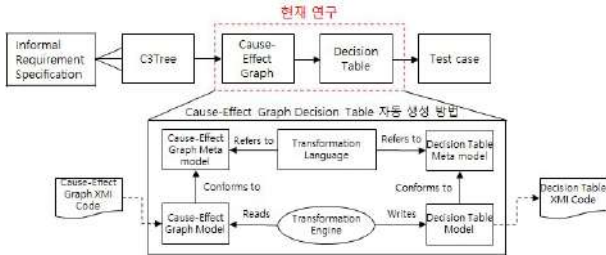
(그림 1) 한국어 요구사항으로부터 테스트 케이스 자동 발생 [3]

메타모델 기반 모델 변환 방법은 Model Driven Architecture(MDA)의 모델 변환 방법이다[6]. 메타모델 기반 변환 엔진은 이기종 레거시 시스템을 상호 운용하고 관리하기 쉽다. 자세한 프로세스는 그림 2와 같다. 변환 엔진은 소스 메타모델과 타겟 메타모델을 참조하여 소스 모델을 타겟 모델로 자동 변환한다.

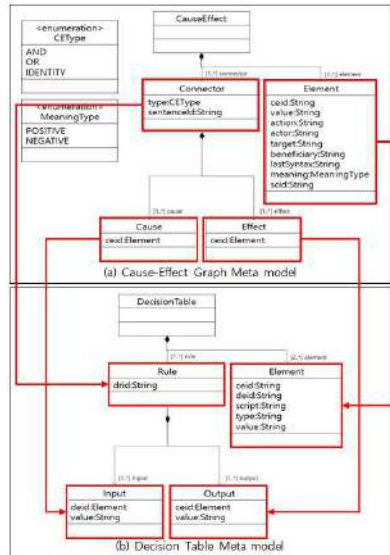


(그림 2) 메타모델 기반의 모델 자동 변환 과정

### 3. 메타모델 기반의 Cause-Effect Graph로부터 Decision Table 자동 발생 과정 설계 및 구현



(그림 3) Cause-Effect Graph로부터 Decision Table 자동 생성 과정



(그림 4) 메타모델 변환 과정

그림 3은 Cause-Effect Graph로부터 Decision Table 자동 생성 과정이다. XMI 소스코드로 구성된 Cause-Effect Graph 모델은 메타모델 변환 엔진에 의해 Decision Table 모델로 자동 변환된다. 이 때 Cause-Effect Graph 모델은 그림 4-(a)의 구성요소를 포함해야 한다. 자동 생성되는 Decision Table 모델은 그림 4-(b)의 구성요소를 포함하여 생성되어야 한다.

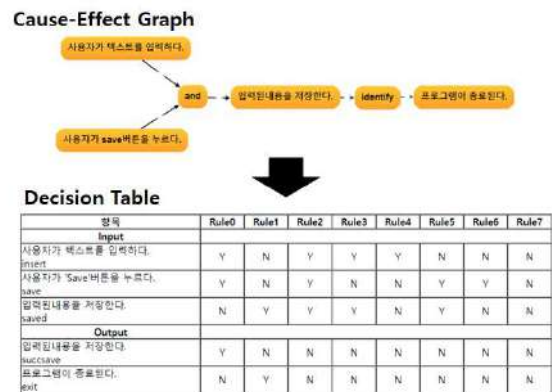
그림 4는 Cause-Effect Graph 메타모델과 Decision Table 메타모델 간의 데이터 구조 변환 룰을 나타낸다. 변환 룰에 의해 Cause-Effect Graph 모델의 구성요소는 Decision Table 모델의 구성요소로 자동 변환된다. 변환룰 코드의 일부는 표 1과 같다.

(표 1) 변환룰 코드의 일부

```
for(int j=0;j<connectorChildren.getLength().length++){
    Node causeEffectNode = connectorChildren.item(j);
    if(causeEffectNode.getNodeType() == Node.ELEMENT_NODE){
        if(causeEffectNode.getNodeName().equals("cause")){
            Element element = outDoc.createElement("element");
            String ceid =
            causeEffectNode.getAttributes().getNamedItem("ceid").getNodeValue();
            String connectionType =
            causeEffectNode.getAttributes().getNamedItem("type").getNodeValue();
            String deid = deidCnt++;
            deidCnt++;
            element.setAttribute("deid", deid);
            element.setAttribute("ceid", ceid);
```

### 4. 실험 결과

그림 5는 구현된 도구가 Cause-Effect Graph를 Decision Table로 자동 변환한 결과이다. Cause-Effect Graph 노드의 모든 조합이 Decision Table 항목으로 변환되었다.



(그림 5) Cause-Effect Graph로부터 자동 생성된 Decision Table

### 5. 결론 및 향후연구

본 논문은 요구사항으로부터 테스트 케이스 자동 생성 과정 중 Cause-Effect Graph로부터 Decision Table 생성 과정을 제안한다. 이 방법은 요구사항으로부터 Decision Table 자동 생성 과정을 위한 추적성을 제공한다.

향후 연구로써, Decision Table로부터 테스트케이스 자동 생성 방법을 연구할 예정이다.

### ACKNOWLEDGEMENT

이 논문은 교육부 및 한국연구재단의 4단계 두뇌한국21 사업의 지원(F21YY8102068)과 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원(No. 2021R1I1A305040711, No. 2021R1I1A1A01044060)을 받아 수행된 연구임.

### 참고문헌

- [1] 권오승, 홍사능, "로그 기반 효과적 반복 테스트", 한국 경영정보학회 2009년 추계학술대회, pp. 685-690, 2009.11.
- [2] Mogyorodi, G.E. Requirements-Based Testing-Cause-Effect Graphing; In Software Testing Services: Ontario, Canada, 2005.
- [3] Woo Sung Jang, R. Young Chul Kim, "Automatic Generation Mechanism of Cause-Effect Graph with

Informal Requirement Specification Based on the Korean Language", Applied Sciences, Vol.11, Issue 24, pp.1-13, 2021.

[4] Myers, G. L., "The Art of Software Testing", Wiley-Interscience, 1979.

[5] Berk Bekiroglu, "A CAUSE-EFFECT GRAPH SOFTWARE TESTING TOOL", European Journal of Computer Science and Information Technology, Vol.5, No.4, pp.11-24, 2017.

[6] OMG-MDA, <https://www.omg.org/mda/>