

【서지사항】

【서류명】	특허출원서
【참조번호】	P230252
【출원구분】	특허출원
【출원인】	
【명칭】	홍익대학교세종캠퍼스산학협력단
【특허고객번호】	1-2013-020039-7
【대리인】	
【명칭】	특허법인태하
【대리인번호】	9-2017-100121-9
【지정된변리사】	황용기
【발명의 국문명칭】	자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템 및 이를 이용한 테스트 케이스 자동 생성 방법
【발명의 영문명칭】	TEST CASE SYSTEM WHICH AUTOMATICALLY GENERATING TEST CASES BASED ON METAMODEL TRANSFORMATION METHOD USING NATURAL LANGUAGE REQUIREMENT AND METHOD FOR AUTOMATICALLY GENERATING TEST CASES USING THE SAME
【발명자】	
【성명】	김영철
【성명의 영문표기】	R. Young Chul Kim
【주소】	경기도 용인시

【발명자】**【성명】** 장우성**【성명의 영문표기】** Jang Woosung**【주민등록번호】** -1XXXXXX**【우편번호】** 30016**【주소】** 세종특별자치시**【출원언어】** 국어**【심사청구】** 청구**【이 발명을 지원한 국가연구개발사업】****【과제고유번호】** 1345362172**【과제번호】** 2021R1I1A3050407**【부처명】** 교육부**【과제관리(전문)기관명】** 한국연구재단**【연구사업명】** 이공학학술연구기반구축**【연구과제명】** NLP BERT Model 기반 자동 리팩토링을 통한 무결점 코드화
연구**【기여율】** 1/1**【과제수행기관명】** 홍익대학교(세종캠퍼스)**【연구기간】** 2021.06.01 ~ 2024.02.29**【취지】** 위와 같이 특허청장에게 제출합니다.

대리인 특허법인태하

(서명 또는 인)

【수수료】

【출원료】	0 면	46,000 원
【가산출원료】	61 면	0 원
【우선권주장료】	0 건	0 원
【심사청구료】	26 항	1,492,000 원
【합계】		1,538,000원
【감면사유】	전담조직(50%감면)[1]	
【감면후 수수료】		769,000 원
【첨부서류】	1. 기타첨부서류[위임장]_1통	

1 : 기타첨부서류

[PDF 파일 첨부](#)

【발명의 설명】

【발명의 명칭】

자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템 및 이를 이용한 테스트 케이스 자동 생성 방법{TEST CASE SYSTEM WHICH AUTOMATICALLY GENERATING TEST CASES BASED ON METAMODEL TRANSFORMATION METHOD USING NATURAL LANGUAGE REQUIREMENT AND METHOD FOR AUTOMATICALLY GENERATING TEST CASES USING THE SAME}

【기술분야】

【0001】 본 발명은 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템 및 이를 이용한 테스트 케이스 자동 생성 방법에 관한 것이다.

【발명의 배경이 되는 기술】

【0003】 현재 소프트웨어 분야에서 자동 테스트 케이스 생성은 매우 중요한 이슈이다. 특히 비정형 요구사항으로부터 테스트 케이스 생성은 다루기 힘든 영역이다.

【0004】 테스트 케이스를 자동 생성할 수 있다면, 소프트웨어 테스트에 대한 비용과 정확도를 개선할 수 있다. 테스트 케이스의 자동 생성에 대해 많은 연구자들이 연구하였고, 소스코드로부터 테스트 케이스 자동 생성 방법은 많이 실용화

되었다. 하지만 소프트웨어 요구사항 비정형화 명세서로부터 테스트 케이스 자동 생성 연구는 아직 미비하다. 특히 자연어 요구사항 기반 명세서로부터 테스트 케이스 자동 생성 방법은 연구가 매우 부족하다.

【0005】 몇몇의 학자들이 해당 분야를 연구 중이지만, 아직까지 자연어 기반 테스트 케이스의 완전한 자동 생성은 어렵다.

【0006】 최근 인공지능 학습 기법을 통한 테스트 케이스 생성 연구가 진행 중이다. 하지만 기존 인공지능 기반 연구는 테스트 케이스 생성 과정의 검증이 어렵다.

【0007】 결과적으로 소프트웨어 테스트의 중요한 요소 중 하나인 요구사항과 테스트 케이스와 추적성 유지가 어렵다.

【0008】 이러한 기존 연구는 테스트 케이스 자동 생성에 대한 연구가 미비하였다. 또한 문장의 부정문, 중첩된 부정문 식별, 접미사의 분리 문제에 대해 고려하지 않았다. 이는 복잡한 문장의 분석 확률을 저해한다.

【0009】 따라서 이와 같은 문제점에 해소하기 위한 대안이 요구되고 있다.

【발명의 내용】

【해결하고자 하는 과제】

【0011】 본 발명은 기존 문제의 해결을 위해 자연어 요구사항을 이용한 메타 모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시

스텝 및 이를 이용한 테스트 케이스 자동 생성 방법을 제공하는 것을 그 목적으로 한다.

【과제의 해결 수단】

【0013】 상술한 목적을 실현하기 위한 본 발명의 일 실시예에 따른 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템은, 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 문장 분석부; 상기 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성하는 원인-결과 그래프 생성부; 상기 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환하는 결정 테이블 생성부; 및 상기 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 상기 입력값 및 결과값 중 적어도 어느 하나가 True인 조합으로 구성되는 테스트 케이스 모델을 생성하는 테스트 케이스 생성부;를 포함할 수 있다.

【0014】 여기서, 상기 문장 분석부는, 상기 자연어 요구사항이 수동 문장인 경우, 이를 능동 문장으로 변화시키는 말뭉치 정규화 과정을 실시할 수 있다.

【0015】 여기서, 상기 문장 분석부는, 상기 C3Tree 모델이 복수로 구성되는 경우, 상기 복수의 C3Tree 모델의 노드에 포함된 형태소의 유사도를 측정하여 형태

소 유사도가 기준값 이상인 경우 중복된 노드들을 병합시킬 수 있다.

【0016】 여기서, 상기 문장 분석부는, 상기 복수의 C3Tree 모델의 노드에 포함된 의미역을 서로 비교하여 의미가 중복되는 노드를 병합시킬 수 있다.

【0017】 여기서, 상기 문장 분석부는, 부정 부사의 포함 여부를 판단하여 상기 부정 부사가 포함된 노드를 부정문으로, 상기 부정 부사가 포함되지 않은 노드를 긍정문으로 판단하여 상기 부정문과 상기 긍정문을 다른 의미로 판단할 수 있다.

【0018】 여기서, 상기 문장 분석부는, 상기 복수의 C3Tree 모델의 노드에 포함된 상기 부정 부사의 개수를 분석하여 홀수 개인 경우 상기 부정문으로, 짝수 개인 경우 상기 긍정문으로 판단할 수 있다.

【0019】 여기서, 상기 문장 분석부는, 상기 복수의 C3Tree 모델의 노드에 접미사가 포함된 동사가 존재하는 경우, 상기 접미사를 제거한 동사를 서로 비교할 수 있다.

【0020】 여기서, 상기 원인-결과 그래프 생성부는, 상기 C3Tree 모델에 포함된 각 절(Clause)을 상기 원인 노드, 상기 결과 노드 및 이들의 연결 노드로 변환할 수 있다.

【0021】 여기서, 상기 결정 테이블 생성부는, 테이블 변환 엔진을 포함하고, 상기 테이블 변환 엔진은, 상기 제1 메타모델 변환 기법을 통해 원인-결과 그래프 메타모델을 참고하여 상기 원인-결과 그래프 모델을 리딩하고, 결정 테이블 변환

룰을 기초로 결정 테이블 메타모델을 참조하여 상기 결정 테이블 모델을 생성할 수 있다.

【0022】 여기서, 상기 테스트 케이스 생성부는, 케이스 변환 엔진을 포함하고, 상기 케이스 변환 엔진은, 상기 제2 메타모델 변환 기법을 통해 상기 결정 테이블 메타모델을 참고하여 상기 결정 테이블 모델을 리딩하고, 테스트 케이스 변환 룰을 기초로 테스트 케이스 메타모델을 참조하여 상기 테스트 케이스 모델을 생성할 수 있다.

【0023】 여기서, 상기 테스트 케이스 모델은, 상기 테스트 케이스 모델을 실행하기 위해 상기 결정 테이블 모델에서 미리 실행되어야 하는 조합을 식별하여 생성되는 사전조건 정보를 포함할 수 있다.

【0024】 여기서, 상기 테스트 케이스 모델을 수신하여 이에 대한 테스트 요청 정보를 생성하고, 이를 타겟 프로그램에 전송 후 상기 테스트 요청 정보에 대한 응답 정보를 수신하는 테스트 실행부를 구비하는 테스트 장치를 더 포함할 수 있다.

【0025】 여기서, 상기 테스트 장치는, 상기 응답 정보를 기설정된 형식에 따라 상기 테스트 케이스 모델의 테스트 결과에 대한 테스트 결과 정보를 생성하는 테스트 결과부를 더 포함할 수 있다.

【0026】 상술한 목적을 실현하기 위한 본 발명의 다른 실시예에 따른 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법은, 문장 분석부를 통해 입

력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계; 원인-결과 그래프 생성부를 통해 상기 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성하는 단계; 결정 테이블 생성부를 통해 상기 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환하는 단계; 및 테스트 케이스 생성부를 통해 상기 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 상기 입력값 및 결과값 중 적어도 어느 하나가 True인 조합으로 구성되는 테스트 케이스 모델을 생성하는 단계;를 포함할 수 있다.

【0027】 여기서, 상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는, 상기 자연어 요구사항이 수동 문장인 경우, 이를 능동 문장으로 변화시키는 말뭉치 정규화 과정을 실시하는 단계를 포함할 수 있다.

【0028】 여기서, 상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는, 상기 C3Tree 모델이 복수로 구성되는 경우, 상기 복수의 C3Tree 모델의 노드에 포함된 형태소의 유사도를 측정하여 형태소 유사도 기준값 이상인 경우 중복된 노드들을 병합시키는 단계를 더 포함할 수 있다.

【0029】 여기서, 상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는, 상기 복수의 C3Tree 모델의 노드에 포함된 의미역을 서로 비교하여 의미가

중복되는 노드를 병합하는 단계를 더 포함할 수 있다.

【0030】 여기서, 상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는, 부정 부사의 포함 여부를 판단하여 상기 부정 부사가 포함된 노드를 부정문으로, 상기 부정 부사가 포함되지 않은 노드를 긍정문으로 판단하여 상기 부정문과 상기 긍정문을 다른 의미로 판단하는 단계를 더 포함할 수 있다.

【0031】 여기서, 상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는, 상기 복수의 C3Tree 모델의 노드에 포함된 상기 부정 부사의 개수를 분석하여 홀수 개인 경우 상기 부정문으로, 짝수 개인 경우 상기 긍정문으로 판단하는 단계를 더 포함할 수 있다.

【0032】 여기서, 상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는, 상기 복수의 C3Tree 모델의 노드에 접미사가 포함된 동사가 존재하는 경우, 상기 접미사를 제거한 동사를 서로 비교하는 단계를 더 포함할 수 있다.

【0033】 여기서, 상기 원인-결과 그래프 생성부를 통해 상기 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성하는 단계는, 상기 C3Tree 모델에 포함된 각 절(Clause)을 상기 원인 노드, 상기 결과 노드 및 이들의 연결 노드로 변환하는 단계를 포함할 수 있다.

【0034】 여기서, 상기 결정 테이블 생성부는, 테이블 변환 엔진을 포함하고, 상기 결정 테이블 생성부를 통해 상기 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환하는 단계는, 상기 테이블 변환 엔진을 통해 상기 제1 메타모델 변환 기법을 통해 원인-결과 그래프 메타모델을 참고하여 상기 원인-결과 그래프 모델을 리딩하고, 결정 테이블 변환 룰을 기초로 결정 테이블 메타모델을 참조하여 상기 결정 테이블 모델을 생성하는 단계를 포함할 수 있다.

【0035】 여기서, 상기 테스트 케이스 생성부는, 케이스 변환 엔진을 포함하고,

【0036】 상기 테스트 케이스 생성부를 통해 상기 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 상기 입력값 및 결과값 중 적어도 어느 하나가 True인 조합으로 구성되는 테스트 케이스 모델을 생성하는 단계는,

【0037】 상기 케이스 변환 엔진을 통해 상기 제2 메타모델 변환 기법을 통해 상기 결정 테이블 메타모델을 참고하여 상기 결정 테이블 모델을 리딩하고, 테스트 케이스 변환 룰을 기초로 테스트 케이스 메타모델을 참조하여 상기 테스트 케이스 모델을 생성하는 단계를 포함할 수 있다.

【0038】 여기서, 상기 테스트 케이스 모델은, 상기 테스트 케이스 모델을 실행하기 위해 상기 결정 테이블 모델에서 미리 실행되어야 하는 조합을 식별하여 생성되는 사전조건 정보를 포함할 수 있다.

【0039】 여기서, 테스트 장치의 테스트 실행부를 통해 상기 테스트 케이스 모델을 수신하여 이에 대한 테스트 요청 정보를 생성하고, 이를 타겟 프로그램에 전송 후 상기 테스트 요청 정보에 대한 응답 정보를 수신하는 단계를 더 포함할 수 있다.

【0040】 여기서, 상기 테스트 장치의 테스트 결과부를 통해 상기 응답 정보를 기설정된 형식에 따라 상기 테스트 케이스 모델의 테스트 결과에 대한 테스트 결과 정보를 생성하는 단계를 더 포함할 수 있다.

【발명의 효과】

【0042】 상술한 구성을 갖는 본 발명의 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템 및 이를 이용한 테스트 케이스 자동 생성 방법에 의하면, 비정형 자연어 요구사항 문장을 단순화하여 이해 관계자 간의 요구사항 이해를 돕고, 테스트 케이스 자동 생성 과정을 통해 테스트 비용을 감소시킬 수 있다.

【도면의 간단한 설명】

【0044】 도 1은 본 발명의 일 실시예에 따른 테스트 케이스 시스템(1000)의 전체적인 동작 방법을 간략히 설명하기 위한 개념도이다.

도 2는 본 발명의 일 실시예에 따른 테스트 케이스 시스템(1000)의 테스트

케이스 장치(100) 및 테스트 장치(300)의 구성을 설명하기 위한 블록도이다.

도 3은 본 발명의 다른 실시예인 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법을 순차적으로 설명하기 위한 흐름도이다.

도 4 내지 도 15는 도 3의 테스트 케이스 생성 방법을 각 단계마다 설명하기 위한 도면이다.

【발명을 실시하기 위한 구체적인 내용】

【0045】 이하, 본 발명의 바람직한 실시예에 따른 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템 및 이를 이용한 테스트 케이스 자동 생성 방법에 대하여 도면을 참조하여 상세히 설명한다. 본 명세서에서는 서로 다른 실시예라도 동일·유사한 구성에 대해서는 동일·유사한 참조번호를 부여하고, 그 설명은 처음 설명으로 갈음한다.

【0047】 도 1은 본 발명의 일 실시예에 따른 테스트 케이스 시스템(1000)의 전체적인 동작 방법을 간략히 설명하기 위한 개념도이다.

【0048】 도시된 바와 같이, 소프트웨어 요구사항 기반 테스트는 소프트웨어 테스트에서 중요한 구성요소이다. 이것은 명료한 비정형 요구사항을 정확하게 분석하고, 목표로 하는 테스트 커버리지를 만족하는 많은 테스트 케이스의 생성이 필요하다.

【0049】 하지만 자연어 기반의 요구사항의 의미를 정의하고 분석하는 것은 매우 어렵다. 또한 자동화가 힘들다. 자연어 요구사항(R) 문장의 복잡도를 줄이고, 간소화 과정이 필요하다.

【0050】 특히 한국에서, 자연어 기반의 요구사항의 복잡도를 줄이고 요구사항 기반의 테스트 케이스 생성 연구는 거의 없다.

【0051】 이에 본 발명은 기존 문제의 해결을 위해 자연어 기반의 요구사항으로부터 테스트 케이스를 자동 생성하기 위한 메커니즘을 제안한다.

【0052】 먼저, 입력 수단을 통하여 자연어 요구사항(R)이 테스트 케이스 장치(100)로 입력될 수 있다. 테스트 케이스 장치(100)는 수신된 자연어 요구사항을 기초로, 비정형 자연어 요구사항(R)을 간소화, 요구사항 문장의 정규화, 요구사항 간소화 과정의 모델화(C3Tree 모델링), 제1 메타모델 변환 기법을 통한 원인-결과 그래프 모델 생성, 제2 메타모델 변환 기법을 통한 테스트 케이스 모델을 생성할 수 있다.

【0053】 이렇게 생성된 테스트 케이스 모델은 테스트 장치(300)를 통해 타켓 프로그램에 테스트 요청 정보로 전송되고, 해당 요청정보에 대한 응답 정보가 수신되면 이를 통해 테스트 결과 정보를 자동으로 생성할 수 있다.

【0054】 이와 같이, 본 발명의 테스트 케이스 시스템(1000)에 의하면, 비정형 한국어 요구사항 문장을 단순화하여 이해 관계자 간의 요구사항 이해를 돕고, 테스트 케이스 자동 생성 과정을 통해 테스트 비용을 감소시킬 수 있다.

【0055】 이상은, 테스트 케이스 시스템(1000)의 간략한 동작 방법에 대하여 설명하였다. 도 2에서는 테스트 케이스 시스템(1000) 구성들에 대하여 구체적으로 설명하도록 한다.

【0057】 도 2는 본 발명의 일 실시예에 따른 테스트 케이스 시스템(1000)의 테스트 케이스 장치(100) 및 테스트 장치(300)의 구성을 설명하기 위한 블록도이다.

【0058】 먼저 테스트 케이스 장치(100)는, 문장 분석부(110), 원인-결과 그래프 생성부(130), 결정 테이블 생성부(150) 및 테스트 케이스 생성부(170)를 포함할 수 있다.

【0059】 문장 분석부(110)는, 입력되는 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현한 C3Tree 모델을 생성하기 위한 수단이다.

【0060】 문장 분석부(110)는, 자연어 요구사항 문장으로부터 원인-결과 그래프를 생성하기 위한 요소들을 식별할 수 있다. 원인-결과 그래프의 요소는 원인 노드, 결과 노드 및 이들 노드 간을 연결하는 연결 노드(AND/OR/NOT)을 포함할 수 있다.

【0061】 이를 위해 문장 분석부(110)는 자연어 요구사항에 관련된 문장이 입력되면, 복문을 중문으로 나누고, 단문들의 관계를 원인, 결과, 연결의 관계로 구분할 수 있다.

【0062】 또한, 문장 분석부(110)는 자연어 요구사항이 수동 문장인 경우, 이를 능동 문장으로 변화시키는 말뭉치 정규화 과정을 실시할 수 있다. 이러한 말뭉치 정규화 과정을 통해 문장이 나뉘는 과정에서 주어가 생략되는 경우 주어를 복구할 수 있다. 결과적으로 세분화된 절로부터 원인, 결과, 연결 관계를 식별할 수 있다.

【0063】 뿐만아니라, 문장 분석부(110)는 C3Tree 모델의 생성 시, C3Tree 모델이 복수로 구성되는 경우, 복수의 C3Tree 모델의 노드에 포함된 형태소의 유사도를 측정하여 형태소 유사도가 기준값 이상인 경우 중복된 노드들을 병합시킬 수 있다. 이는 자카드 유사도(Jaccard similarity) 방식을 적용하여 형태소 유사도를 산출할 수 있다.

【0064】 이때, 자카드 유사도가 기준값(ex. 0.8) 이상이라도 문장 간의 의미가 다를 수 있다. 이를 위해 문장 분석부(110)는 복수의 C3Tree 모델의 노드에 포함된 의미역을 서로 비교하여 의미가 중복되는 노드를 병합하거나, 서로 다른 문장으로 인식할 수 있다.

【0065】 이에 더하여, 기존 연구에는 자연어 요구사항에 관한 문장에 부정문이 포함되거나 접미사가 포함되는 경우 문장 비교의 결과가 부정확하게 도출되는 문제점이 존재하였다. 이에, 본 발명의 문장 분석부(110)는 해당 문제 해결을 위해 부정문의 비교, 중첩된 부정문의 식별, 접미사가 포함된 형태소의 비교를 적용할 수 있다.

【0066】 먼저, 문장 분석부(110)는 자연어 요구사항에 관한 문장의 노드를

분석하여 부정 부사의 포함 여부를 판단할 수 있다. 이때, 부정 부사가 포함된 노드는 부정문으로, 부정 부사가 포함되지 않은 노드는 긍정문으로 판단하고, 부정문과 긍정문을 서로 다른 의미로 판단할 수 있다. 또한, 복수의 C3Tree 모델의 노드에 포함된 부정 부사의 개수를 분석하여 홀수 개인 경우 해당 문장을 부정문으로, 짝수 개인 경우 해당 문장을 긍정문으로 판단할 수 있다. 이를 통해 자연어 요구사항 문장에 대한 유사도를 보다 명확하게 구분할 수 있다.

【0067】 다만, 문장 분석부(110)는 복수의 C3Tree 모델의 노드에 접미사가 포함된 동사가 존재하는 경우, 해당 접미사를 제거한 동사를 서로 비교하여 유사도를 판단할 수 있다. 이는, 접미사에 의해 동사가 서로 다른 의미로 분석되는 것을 방지하기 위한 것이다. 이에 대한 구체적인 예시는 후술하도록 한다.

【0068】 원인-결과 그래프 생성부(130)는 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성할 수 있다. 이는 문장 분석부(110)를 통해 복수의 C3Tree 모델이 통합된 이후, C3Tree 모델의 구성요소들을 원인-결과 그래프 모델의 구성요소로 변환시키는 방식으로 이루어질 수 있다. 구체적으로 원인-결과 그래프 생성부(130)는 C3Tree 모델에 포함된 각 절(Clause)을 원인 노드, 결과 노드 및 이들의 연결 노드로 변환시켜 원인-결과 그래프 모델을 생성할 수 있다.

【0069】 결정 테이블 생성부(150)는, 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를, 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환할 수 있다.

【0070】 이를 위해 결정 테이블 모델은 테이블 변환 엔진을 포함하며, 제1

메타모델 변환 기법을 통해, 원인-결과 그래프 메타모델을 참고하여 원인-결과 그래프 모델을 리딩하고, 결정 테이블 변환 룰을 기초로 결정 테이블 메타모델을 참조하여 결정 테이블 모델을 생성할 수 있다. 이에 대한 구체적인 예시는 후술하도록 한다.

【0071】 테스트 케이스 생성부(170)는, 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 입력값과 결과값이 True인 조합으로 구성되는 테스트 케이스 모델을 생성할 수 있다. 이를 위해 테스트 케이스 생성부(170)는 케이스 변환 엔진을 포함하며, 제2 메타모델 변환 기법을 통해 결정 테이블 메타모델을 참고하여 결정 테이블 모델을 리딩하고, 테스트 케이스 변환 룰을 기초로 테스트 케이스 메타모델을 참조하여 테스트 케이스 모델을 생성할 수 있다. 이에 대한 구체적인 예시는 후술하도록 한다.

【0072】 이렇게 생성되는 테스트 케이스 모델은 테스트 장치(300)를 통해 실행되어 검증을 실시할 수 있다.

【0073】 이를 위한 테스트 장치(300)는, 테스트 실행부(310) 및 테스트 결과부(330)를 포함할 수 있다.

【0074】 테스트 실행부(310)는, 테스트 케이스 모델을 수신하여 이에 대한 테스트 요청 정보를 생성하고, 이를 타겟 프로그램에 전송 후 테스트 요청 정보에 대한 응답 정보를 수신할 수 있다. 이때, 테스트 스크립트 생성을 위한 테스트 커멘트는 상술한 결정 테이블에 삽입될 수 있으며 이 정보는 결정 테이블 메타모델에서 Element 클래스 내 스크립트에 저장될 수 있다. 따라서, 테스트 실행부(310)는

테스트 수행 시 타겟 프로그램과 TCP/IP 통신을 통해 해당 스크립트를 상호전송할 수 있다.

【0075】 테스트 결과부(330)는, 타겟 프로그램으로부터 수신된 응답 정보를 기준값과 비교하여 테스트 케이스 모델의 테스트 결과에 관한 테스트 결과 정보를 생성할 수 있다. 이렇게 생성된 테스트 결과 정보는 도시하지 않은 출력부를 통해 출력되거나, 통신부를 통해 관리자 단말기나 서버 등으로 전송될 수 있다.

【0076】 이와 같은 테스트 케이스 시스템(1000)에 의하면 한국어 문장과 같은 비정형 자연어 요구사항에 대하여 구분, 식별, 중복 제거, 정규화, 통합 과정 등을 실시하고, 해당 결과의 유효성 여부를 테스트까지 자동으로 실시함에 따라 전체 시스템의 신뢰성, 정확도 향상은 물론 시스템 운용의 편리성까지 보다 향상시킬 수 있다.

【0077】 이상은 테스트 케이스 시스템(1000)의 구성들에 대하여 살펴보았다. 이하에서는 해당 시스템의 구동 방법을 각 단계 별 예시를 통해 보다 구체적으로 설명하도록 한다.

【0079】 도 3은 본 발명의 다른 실시예인 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법을 순차적으로 설명하기 위한 흐름도이고, 도 4 내지 도 15는 도 3의 테스트 케이스 생성 방법을 각 단계마다 설명하기 위한 도면이다.

【0080】 본 실시예에서는 테스트 케이스 시스템의 동작 방법을 예시적인 도면과 같이 순차적으로 설명하도록 하며, 상술한 구성 간의 구체적인 설명 및 부호에 대한 중복 기재는 생략하도록 한다.

【0081】 먼저 입력 수단을 통해 테스트 케이스 장치로 자연어 요구사항이 입력될 수 있다(S11).

【0082】 도 4(a)는 문장 분석부를 통해 자연어 요구사항으로 구성된 문장으로부터 원인절과 결과절을 식별하는 예시이다. 도 4(a)의 위의 문장이 입력된 자연어 요구사항에 관한 원본 문장이고, 밑의 두 개의 문장이 원본 문장에서 원인절(원인)과 결과절(결과)이 구분된 결과이다.

【0083】 일반적인 경우, 한국어로 구성되는 자연어 요구사항인 문장은 절을 구분하기 위해 연결어미(連結語尾)를 사용한다. 연결어미는 두 절을 연결하는 형태소이며, ‘-고’, ‘-면’ 등으로 구성된다. 연결어미 중 하나인 조건 연결어미는 조건절과 결과절을 연결하는 형태소다.

【0084】 본 실시예에서는 자연어 요구사항에서 조건 연결어미 형태소들을 식별하여 조건절과 결과절을 구분한다. 조건 연결어미의 종류는 도 4(b)와 같고, 조건 연결어미의 복합 형태는 도 4(c)와 같다. 이 외에도 예외상황에서 조건 연결어미를 식별하기 위한 예외 사전을 구성할 수 있다.

【0085】 조건 연결어미가 식별되었다면, 문장 분석부는 이것을 긍정적 조건 연결어미와 부정적 조건 연결어미로 구분할 수 있다.

【0086】 긍정적 조건 연결어미로 연결된 절들은 원인-결과(if-then) 관계로 식별된다. 부정적 조건 연결어미로 연결된 절들은 원인-NOT결과(if-then not) 관계로 식별된다.

【0087】 도 5(a)는 자연어 요구사항의 문장으로부터 연결 관계 중 AND 관계를 식별하는 예시이고, 도 5(b)는 OR 관계를 식별하는 예이다. 문장에서 접속 연결어미를 식별하여 접속절과 후행절을 구분할 수 있다. 접속 연결어미의 종류 중 일부는 도 5(c)와 같다.

【0088】 문장 분석부는 접속 연결어미의 형태소를 식별한 경우, 연결 관계를 AND 관계와 OR 관계로 구분할 수 있다.

【0089】 자연어 요구사항에 관한 문장이 원인, 결과, 연결(AND, OR, NOT) 관계로 구분되는 경우, 문장 분석부는 말뭉치 정규화 과정을 실시할 수 있다. 말뭉치 정규화는 수동 문장을 능동 문장으로 변환하고, 절을 나누는 과정에서 사라진 주어를 복구할 수 있다. 도 6(a)는 말뭉치 정규화 과정 중 수동 문장을 능동 문장으로 변환한 예시이다.

【0090】 도 6(b)는 제1 자연어 요구사항(R1)을 나타낸다.

【0091】 도 6(c)는 (R1)의 간소화 과정을 보여준다. 문장 분석부는 (a)제1 자연어 요구사항(R1)의 문장에서 조건절을 식별할 수 있다. 이 후 (b)조건절 내의 접속절을 식별할 수 있다. 다음 (c-1)조건절 내의 후행절을 식별 할 수 있다. (c-2)식별된 후행절을 말뭉치 정규화하여 주어를 복구할 수 있다. (d)조건절에 대한

결과절을 식별할 수 있다. (e) 수동 문장을 능동 문장으로 변환한다.

【0092】 결과적으로 제1 자연어 요구사항(R1) 문장은 세 개의 간소화된 문장으로 나뉠 수 있다. 첫 번째 문장과 두 번째 문장은 AND 관계를 가진다. 첫 번째, 두 번째 문장은 원인이 된다. 세 번째 문장은 결과가 될 수 있다.

【0093】 이와 같은 자연어 요구사항에 대한 식별을 실시한 문장 분석부는 이를 토대로 C3Tree 모델을 생성할 수 있다(S13).

【0094】 이는 문장 분석부의 자연어 요구사항에 대한 식별 과정을 C3Tree 모델(Conditional and Conjunction Clause Tree Model)로 표현하는 것이다.

【0095】 C3Tree 모델은 앞선 자연어 요구사항 문장의 절 분리 과정을 이진 트리 형태로 표현할 수 있다.

【0096】 C3Tree 모델의 노드의 종류는 도 7(a)와 같다.

【0097】 <<Sentence>> 노드는 복잡한 원본 문장이다. <<Complex-Clause>> 노드는 간소화되고 있는 중간 노드로써, 여러 <<Cause>> 노드를 포함한다. <<Cause>> 노드는 말단 노드로써, 문장의 단순화가 완료된 노드이다.

【0098】 C3Tree 모델의 연관 관계는 도 7(b)와 같다.

【0099】 COND-P는 자식 노드가 긍정적인 조건 관계이다. COND-N는 자식 노드가 부정적인 조건 관계이다. CONJ-AND는 자식 노드가 AND 관계이다. CONJ-OR는 자식 노드가 OR 관계이다.

【0100】 도 7(c)는 도 7(a)를 C3Tree 노드로 표현한 결과이다. 문장 분석부에 의한 제1 자연어 요구사항의 C3Tree 모델 생성 순서는 다음과 같다.

【0101】 (a) 제1 자연어 요구사항에서 원인절과 결과절을 식별한다. (b) 원본 문장을 <<Sentence>> 노드로 표현한다. (c) 원인절을 <<Complex-Clause>> 노드로 표현한다. (d) 결과절/접속절/후행절을 <<Clause>> 노드로 표현한다. (e) 각 노드 간의 연관관계를 표현한다.

【0102】 상기와 같은 과정을 통해 문장 분석부는 자연어 요구사항(제1 자연어 요구사항(R1))을 기초로하여 이에 대응되는 C3Tree 모델을 생성할 수 있다.

【0103】 하나의 자연어 요구사항을 갖는 문장은 하나의 C3Tree 모델로 생성될 수 있다. 따라서, 복수의 자연어 요구사항이 입력되는 경우 이에 대응되는 복수의 C3Tree 모델들이 생성되며 이때, 문장 분석부는 복수의 C3Tree 모델들의 중복을 제거하여 하나로 통합할 수 있다(S15). 즉, 중복되는 의미를 가지는 말단 노드들은 통합될 수 있다.

【0104】 기존 연구에서는 복수의 C3Tree 모델의 중복을 제거하기 위해 자카드 유사도와 의미역을 비교하여 노드 간의 유사도를 측정하였다.

【0105】 자카드 유사도는 A집합과 B집합 간의 유사도를 측정하여 0 ~ 1.0 사이의 값으로 나타낸다. 자카드 유사도의 공식은 도 8a와 같다. 본 실시예에서의 문장 분석부는 자카드 유사도를 사용하여 C3Tree 모델 내 각 노드들의 형태소 유사도를 측정한다.

【0106】 도 8b는 제2 자연어 요구사항(R2)의 문장을 보여준다. 도 8c는 제1 자연어 요구사항(R1)의 말단 노드 중 하나와 제2 자연어 요구사항(R2)의 말단 노드 중 하나의 형태소를 통해 자카드 유사도를 측정한다.

【0107】 형태소 비교 시에는 조사를 제거하고, 종결어미의 받침을 제거한 후 비교할 수 있다. 제1 자연어 요구사항(R1)에서는 조사 '이', '를'이 제거되었고, 제2 자연어 요구사항(R2)에서는 종결어미의 받침 '니'이 제거되었다. 결과적으로 박스로 표시되는 노드 간의 자카드 유사도는 1.0이다.

【0108】 일반적으로 자카드 유사도 0.8을 기준값으로 하여 그 이상인 경우 문장 간의 의미가 유사하다고 판단할 수 있다. 하지만 노드 간 자카드 유사도가 기준값(0.8) 이상이라도 문장 간의 의미가 다를 수 있다.

【0109】 이 문제의 해결을 위해 문장 분석부는 추가적으로 의미역을 식별하여 의미가 중복되는 노드를 결합시킬 수 있다. 의미역은 술어와 관련된 의미론적 구조에서 참가자의 역할을 나타낸다. 본 실시예에서는 한국전자통신연구원(ETRI)의 Exobrain 라이브러리를 통해 의미역을 분석할 수 있다. 도 9a는 Exobrain 라이브러리에 적용된 세종전자사전 의미역의 일부이다.

【0110】 도 9b는 제1,2 자연어 요구사항(R1, R2)의 박스로 표시되는 노드 간의 의미역 유사도이다. 의미역이 모두 동일하다.

【0111】 하지만, 기존 연구에서는 문장에 부정문이 포함되었거나, 접미사가 포함되었다면 문장 비교를 부정확하게 수행되는 문제점이 존재하였다.

【0112】 이에 본 실시예에서는 상술한 문제를 해소하기 위해 문장 분석부를 통해 부정문 비교, 중첩된 부정문 식별, 접미사가 포함된 형태소의 비교 방법을 적용할 수 있다.

【0113】 문장에 부정 부사가 포함된 경우 해당 문장은 부정적인 의미인 부정문으로 판단될 수 있다. 예컨대, “공부를 하다.”와 “공부를 안하다”는 유사한 형태소를 가지지만, 반대의 의미를 가진다. 그렇기 때문에 부정문은 자카드 유사도가 높더라도 다른 의미로 식별되어야 한다. 이에 문장 분석부는, 제1 자연어 요구사항 및 제2 자연어 요구사항의 자카드 유사도가 기준값 이상인 경우에도, 부정 부사의 존재 여부를 판단하여 이들간의 유사도 여부를 판단할 수 있다.

【0114】 또한 문장은 여러 개의 부정적 의미를 중첩해서 쓸 수 있다. 대부분의 경우 두 번의 부정이 중첩되면 긍정의 의미로 해석된다. 결과적으로, 문장 분석부는 부정 부사의 개수를 분석하여 홀수 개인 경우 부정문으로, 짝수 개인 경우 긍정문으로 판단할 수 있다.

【0115】 또한, 문장 분석부는 문장 내 동사에서 접미사를 제거한다. 예를 들어, "열리-"와 같은 동사는 동사와 사동접미사가 결합된 형태를 가진다. 사동 접미사는 동사를 사동적 의미로 변환시키는 역할을 한다. 사동 접미사는 형태소 분석기에서 자동 분석되지 않는다. “열리다”와 “열다”를 형태소 분석기는 다른 형태소로 인식하기 때문에 실제적인 의미는 같지만 자카드 유사도에서 차이가 발생할 수 있다.

【0116】 이 경우 문장 분석부는 “열리” (동사)+ “다” (종결어미)에서 "열"을

동사로 식별하고 "리"를 접미사로 식별하여 서로 분리한 후 "리"를 삭제하고 최종적으로 “열다”로 만들 수 있다.

【0117】 결과적으로 문장 분석부는 주어, 동사, 목적어와 같은 중요 형태소들만 비교하여 기준값(0.8) 이상의 유사도를 가지는 문장은 유사한 형태소를 가진다고 판단한다.

【0118】 도 9c는 제1 자연어 요구사항(R1)과 제2 자연어 요구사항(R2)의 C3Tree 모델의 중복이 제거되어 통합된 결과를 보여준다. 한 개의 노드가 통합되었다.

【0119】 이렇게 C3Tree 모델의 중복이 제거되어 통합되면 원인-결과 그래프 생성부를 통해 통합된 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성할 수 있다(S17).

【0120】 원인-결과 그래프 생성부는, 통합된 C3Tree 모델의 구성요소들을 원인-결과 그래프 모델의 구성요소로 변환시켜 원인-결과 그래프 모델을 생성할 수 있다.

【0121】 도 10a는 C3Tree 모델과 이를 변환한 원인-결과 그래프 모델 간의 연관 관계를 나타낸다. 예컨대, C3Tree 모델의 <<Clause>>는 원인-결과 그래프 모델의 Node로 변환된다.

【0122】 도 10b는 통합된 C3Tree 모델로부터 원인-결과 그래프 모델을 생성한 예이다. 원인-결과 그래프 생성부는 C3Tree 모델의 최하위 레벨에 있는 4개의

<<Clause>> 노드를 원인-결과 그래프 모델의의 노드 A~D로 변환시킬 수 있다.

【0123】 도시된 바와 같이, COND-P의 왼쪽 노드는 원인 노드로 변환되고, COND-P의 오른쪽 노드는 결과 노드로 변환되며, CONJ-AND는 AND로 변환될 수 있다.

【0124】 이렇게 원인-결과 그래프 모델을 생성한 뒤, 결정 테이블 생성부를 통해 원인-결과 그래프 모델을 결정 테이블 모델로 변환시킬 수 있다(S19).

【0125】 이를 위해, 결정 테이블 모델은 제1 메타모델 변환 기법인 MDA(Model Driven Architecture)을 적용할 수 있다. 이는, 테이블 변환 룰의 용이한 추가, 수정, 삭제 기능을 지원할 수 있다.

【0126】 도 11은 원인-결과 그래프 모델의 메타모델을 설계하고, 원인-결과 그래프 메타모델을 기반으로 원인-결과 그래프 모델 예제를 생성한 예시이다. 화살표를 사용하여 원인-결과 그래프 메타모델, 원인-결과 그래프 모델의 XMI 코드, 실제 원인-결과 그래프 모델 간의 관계를 나타낸다.

【0127】 도 11의 상단은 원인-결과 메타모델의 구조이다. CauseEffect, Connector, Element, Cause, Effect 클래스로 구성될 수 있다.

【0128】 CauseEffect는 여러 개의 Connector와 Element 정보를 포함할 수 있다.

【0129】 Element는 Cause-Effect Graph 내 노드들의 정보를 저장할 수 있다.

【0130】 ceid는 Element의 고유번호이다.

【0131】 alue는 자연어 문장이다.

【0132】 action, actor, target, beneficiary는 문장의 의미역이다.

【0133】 lastSyntax는 구문분석에서 문장의 마지막 동사이다.

【0134】 Meaning은 문장의 NOT 여부이다.

【0135】 scid는 원본 문장의 고유번호이다.

【0136】 Connector는 연결관계를 가지는 Cause(원인)와 Effect(결과)를 포함한다.

【0137】 Cause는 Cause에 해당되는 Element 정보를 저장한다.

【0138】 ceid는 Element의 번호이다.

【0139】 Type은 CETYPE 중 하나이다.

【0140】 AND는 다음 노드와 AND 관계임을 의미한다.

【0141】 OR는 다음 노드와 OR 관계임을 의미한다.

【0142】 LAST는 마지막 Cause임을 의미한다.

【0143】 Effect는 Effect에 해당되는 Element 정보를 저장한다.

【0144】 ceid는 Element의 번호이다.

【0145】 Type은 CETYPE 중 하나이다.

【0146】 ffect는 CETYPE의 속성 값으로 LAST만 사용한다.

【0147】 도 11의 중간은 원인-결과 그래프 메타모델을 기반으로 원인-결과 그래프 모델의 XMI 코드를 생성한 예이다. 상단과 중간 사이의 화살표는 원인-결과 메타모델과 XMI 코드 간의 매핑 정보이다. CauseEffectGraph태그는 4개의 element

태그와 2개의 connector 태그를 포함한다.

【0148】 각 connector 태그는 n개의 cause 태그와 1개의 effect 태그를 포함한다.

【0149】 cause 태그는 아래의 cause 태그와의 AND 또는 OR 관계 정보를 포함한다.

【0150】 도 11의 하단은 XMI 코드를 원인-결과 그래프 모델로 표현한 결과이다.

【0151】 네 개의 element 태그는 노드로 표현된다.

【0152】 2개의 connector 태그는 원인-결과노드 쌍으로 표현된다.

【0153】 cause 태그는 원인 노드로 표현되고, effect 태그는 결과 노드로 표현된다.

【0154】 도 12는 제1 메타모델 변환 기법을 통해 원인-결과 그래프 모델을 결정 테이블 모델로 변환하는 과정이다.

【0155】 중간의 테이블 변환 엔진은 원인-결과 그래프 메타모델을 참조하여 생성된 원인-결과 그래프 모델을 읽고, 테이블 변환 엔진을 실행하여 결정 테이블 메타모델을 참조한 결정 테이블 모델을 생성할 수 있다.

【0156】 테이블 변환 엔진은 프로그램일 수 있다.

【0157】 원인-결과 그래프 메타모델, 원인-결과 그래프 모델, 결정 테이블 메타모델 및 결정 테이블 모델을 XMI 파일일 수 있다.

【0158】 테이블 변환 룰은 ATLAS Transformation Language(ATL) 파일일 수 있다.

【0159】 도 12a의 우측 상단 모델은 결정 테이블 메타모델의 구조이다. 결정 테이블 메타모델은 DecisionTable, Rule, Element, Input, Output 클래스를 포함할 수 있다.

【0160】 DecisionTable은 여러개의 Rule, Element를 포함할 수 있다.

【0161】 Element는 Decision Table 내 항목들의 정보를 저장할 수 있다.

【0162】 ceid는 Causes-Effect Graph 내 Element의 고유번호일 수 있다.

【0163】 deid는 Input 또는 Output 항목의 고유번호일 수 있다.

【0164】 Script는 테스트 커맨드 정보를 저장할 수 있다.

【0165】 Type은 Element의 속성으로써 input 또는 output의 값을 가질 수 있다.

【0166】 value는 자연어 문장일 수 있다.

【0167】 Rule은 Input과 Output의 조합 정보를 가질 수 있다.

【0168】 Input은 Decision Table의 Input 항목들의 정보를 저장할 수 있다.

【0169】 deid는 타겟이 되는 Element의 고유번호를 저장할 수 있다.

【0170】 value는 해당 Input 항목의 값의 true인지 false인지 저장하는 정보를 가질 수 있다.

【0171】 True는 Y가 저장되고, False는 N이 저장될 수 있다.

【0172】 Output은 Decision Table의 Output 항목들의 정보를 저장할 수 있다. 속성에 대한 설정은 Input과 같을 수 있다.

【0173】 도 12a의 중앙 상단의 화살표는 원인-결과 그래프 메타모델과, 결정 테이블 메타모델 간의 변환 룰이다.

【0174】 Connector는 Rule로 변환될 수 있다.

【0175】 Rule의 개수는 2의 Cause 개수 제곱일 수 있다.

【0176】 Element는 Element로 변환될 수 있다.

【0177】 Cause는 Input으로 변환될 수 있다.

【0178】 Rule에 포함되는 In-put의 개수는 모든 Cause들을 합친 개수일 수 있다.

【0179】 Rule에 포함되는 Effect는 Output으로 변환될 수 있다.

【0180】 Output의 개수는 모든 Effect들을 합친 개수일 수 있다.

【0181】 도 12a의 하단은 도 11의 원인-결과 그래프 모델이 결정 테이블 모델로 변환될 결과이다. 모든 Cause(원인) 노드는 Input(입력값)으로 변환되었고, 모든 Effect(결과) 노드는 Output(결과값)으로 변환되었다. 그리고 각 Input과 Output의 True/False 조합을 Rule로 나열하였다.

【0182】 도 12b는 도 11의 원인-결과 그래프 XMI 코드가 테이블 변환 엔진에 의해 결정 테이블 XMI 코드로 자동 변환된 결과의 일부이다.

【0183】 이렇게 결정 테이블 생성부를 통해 결정 테이블 모델이 생성되는 경우, 결정 테이블 생성부는 추가로 테스트 스크립트 생성을 위한 테스트 커맨드를 결정 테이블 모델에 삽입할 수 있다. 이 정보는 결정 테이블 메타모델에서 Element 클래스 내 script에 저장될 수 있다. 이 후 단계에서 테스트 케이스 모델이 완성되고, 테스트를 수행할 경우 테스트 장치는 TCP/IP 통신을 통해 타겟 프로그램과 script를 상호전송할 수 있다.

【0184】 Element의 type이 input인 경우, 테스트 장치가 타겟 프로그램에게 테스트 요청 정보인 script를 전달한다. 반면 Element의 type이 output인 경우, 타겟 프로그램이 테스트 요청 정보에 대한 응답 정보인 script를 테스트 장치에 전달한다.

【0185】 도 13a는 도 12b 결정 테이블 모델의 XMI 코드에 script가 삽입된 결과를 나타낸다.

【0186】 도 13b는 script가 삽입된 결정 테이블 모델을 보여준다. 문장의 아래에 script를 출력된다.

【0187】 이와 같이 결정 테이블 모델이 생성되면 테스트 케이스 생성부는 결정 테이블 테스트 케이스 모델로 변환할 수 있다(S19).

【0188】 도 14a는 제2 메타모델 변환 기법을 통해 결정 테이블 모델을 테스트 케이스 모델로 변환하는 과정이다.

【0189】 케이스 변환 엔진은 결정 테이블 메타모델을 참조하여 테스트 케이스 변환 룰을 기초로 테스트 케이스 메타모델을 참조하여 테스트 케이스 모델을 생성할 수 있다.

【0190】 테스트 케이스 메타모델은 Testcase, Case, Precondition(사전조건 정보), Send(전송 정보), Recv(수신 정보) 클래스를 포함할 수 있다.

【0191】 Testcase는 여러 개의 Case 정보를 포함할 수 있다.

【0192】 Case는 테스트 케이스 모델의 여러 개의 Precondition, Send, Recv를 포함할 수 있다.

【0193】 drid 속성은 테스트 케이스 모델 내 Element의 고유번호이다.

【0194】 Precondition은 테스트 케이스 모델의 실행을 위한 사전조건 정보이다. 결정 테이블 모델에서 미리 실행되어야 하는 조합을 식별하여 생성할 수 있다.

【0195】 Precondition은 결정 테이블 모델 내 항목 중 하나의 원인 노드 및 결과 노드로부터 발생된 Input(입력값)과 Output(결과값)이 있고, 이 항목의 Input의 값이 true인 경우, 해당 항목의 Output을 true로 만드는 테스트 케이스 변환 룰의 Input 조합으로 만들어질 수 있다. Input과 해당 Element의 script는 Send로 변환된다. Output과 해당 Element의 script는 Recv로 변환된다.

【0196】 script는 타겟 프로그램에게 전달되는 script 코드일 수 있다. script는 결정 테이블 모델에서 입력값과 결과값이 True인 케이스의 개별 조합으로 각각 생성될 수 있다.

【0197】 value는 자연어 문장이다.

【0198】 Recv는 테스트 장치가 타켓 프로그램으로부터 전달받는 정보일 수 있다. 각 속성은 Precondition과 같다.

【0199】 도 14a의 중앙 상단은 결정 테이블 메타모델과 테스트 케이스 메타모델 간의 변환 과정에 관한 것이다. Rule은 Case로 변환될 수 있다.

【0200】 도 14a의 하단은 도 11a의 결정 테이블 모델을 테스트 케이스 모델로 변환한 결과이다. 모든 룰이 테스트 케이스 모델로 변환된다. 이때, 모든 조합이 False인 룰은 변환되지 않는다.

【0201】 도 14b는 도 12a의 거절 테이블 모델에 대한 테스트 케이스 모델의 XMI 코드를 생성한 예이다.

【0202】 도 14a의 하단 에서처럼 XMI 코드를 테스트 케이스 모델로 가시화한 경우, 테스트 케이스 모델은 7개의 case 태그를 포함할 수 있다. 이때, Case는 n개의 precondition, send, recv 정보를 포함할 수 있다.

【0203】 이렇게 테스트 케이스 모델이 생성되면 테스트 장치는 이를 수신하여 테스트 실행부를 통해 이에 대한 테스트 요청 정보를 생성하고, 이를 타켓 프로그램으로 전송할 수 있다(S23).

【0204】 이는 자동적으로 동작으로 테스트 실행부는 테스트 케이스 모델에 포함되는 script를 타켓 프로그램에 전달하여 실행되도록 할 수 있다.

【0205】 도 15a는 테스트 장치가 테스트 프로그램인 타겟 프로그램에 테스트 요청 정보인 script를 전달하는 과정이다.

【0206】 테스트 요청 정보는, 테스트 장치를 통해 내부에서 생성된 스크립트 코드가 테스트 케이스 모델의 순서에 맞춰진 상태로 타겟 프로그램에 전달될 수 있다.

【0207】 이렇게 테스트 요청 정보가 전송된 이 후, 각 테스트 케이스 모델의 응답 스크립트인 테스트 결과에 관한 응답 정보가 타겟 프로그램으로부터 테스트 장치로 전달될 수 있다.

【0208】 응답 정보가 수신되면 테스트 장치는 테스트 결과부를 통해 응답 정보를 기설정된 형식에 따라 테스트 케이스 모델의 테스트 결과에 관한 테스트 결과 정보를 생성할 수 있다(S25). 이렇게 생성된 테스트 결과 정보는 도시하지 않은 디스플레이부를 통해 출력하거나, 다른 외부 단말기 또는 서버 등으로 전송될 수 있다.

【0209】 도 15b는 테스트 결과 정보에 관한 것이다. 테스트 결과 정보 도는 XMI 코드로 구현될 수 있다.

【0210】 Test Steps는 테스트 항목일 수 있다.

【0211】 Test Send Script는 타겟 프로그램에게 전달된 스크립트일 수 있다.

【0212】 Expected Result는 예상 결과일 수 있다.

【0213】 Expected Receive Script는 타켓 프로그램으로부터 전달받는 예상 스크립트일 수 있다.

【0214】 Actual Receive Script는 타켓 프로그램으로부터 전달받은 실제 스크립트일 수 있다.

【0215】 Status는 테스트 케이스의 판정 여부일 수 있다.

【0216】 Notes는 기타 사항을 적는 항목일 수 있다.

【0217】 이와 같은, 테스트 케이스 시스템 이용한 테스트 케이스 생성 방법에 의하면 한국어와 같은 자연어 요구사항의 입력 시, C3Tree 모델, 원인-결과 그래프 모델, 결정 테이블 모델, 테스트 케이스 모델로 순서로 자동 변환하여, 자연어 요구사항의 명확성을 보다 향상시키고, 테스트 장치를 통해 검증을 실시함으로써 생성된 테스트 케이스 모델의 신뢰성을 보다 향상시킬 수 있다.

【0219】 상기와 같은 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템 및 이를 이용한 테스트 케이스 자동 생성 방법은, 위에서 설명된 실시예들의 구성과 작동 방식에 한정되는 것이 아니다. 상기 실시예들은 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 다양한 변형이 이루어질 수 있도록 구성될 수 있다.

【부호의 설명】

【0221】 1000: 테스트 케이스 시스템 170: 테스트 케이스 생성부

100: 테스트 케이스 장치 300: 테스트 장치

110: 문장 분석부 310: 테스트 실행부

130: 원인-결과 그래프 생성부 330: 테스트 결과부

150: 결정 테이블 생성부

【청구범위】

【청구항 1】

입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 문장 분석부;

상기 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성하는 원인-결과 그래프 생성부;

상기 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환하는 결정 테이블 생성부; 및

상기 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 상기 입력값 및 결과값 중 적어도 어느 하나가 True인 조합으로 구성되는 테스트 케이스 모델을 생성하는 테스트 케이스 생성부;를 포함하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 2】

제1항에 있어서,

상기 문장 분석부는,

상기 자연어 요구사항이 수동 문장인 경우, 이를 능동 문장으로 변화시키는 말뭉치 정규화 과정을 실시하는, 자연어 요구사항을 이용한 메타모델 변환 방법을

기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 3】

제2항에 있어서,

상기 문장 분석부는,

상기 C3Tree 모델이 복수로 구성되는 경우, 상기 복수의 C3Tree 모델의 노드에 포함된 형태소의 유사도를 측정하여 형태소 유사도가 기준값 이상인 경우 중복된 노드들을 병합시키는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 4】

제3항에 있어서,

상기 문장 분석부는,

상기 복수의 C3Tree 모델의 노드에 포함된 의미역을 서로 비교하여 의미가 중복되는 노드를 병합하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 5】

제4항에 있어서,

상기 문장 분석부는,

부정 부사의 포함 여부를 판단하여 상기 부정 부사가 포함된 노드를 부정문으로, 상기 부정 부사가 포함되지 않은 노드를 긍정문으로 판단하여 상기 부정문과 상기 긍정문을 다른 의미로 판단하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 6】

제5항에 있어서,

상기 문장 분석부는,

상기 복수의 C3Tree 모델의 노드에 포함된 상기 부정 부사의 개수를 분석하여 홀수 개인 경우 상기 부정문으로, 짝수 개인 경우 상기 긍정문으로 판단하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 7】

제6항에 있어서,

상기 문장 분석부는,

상기 복수의 C3Tree 모델의 노드에 접미사가 포함된 동사가 존재하는 경우, 상기 접미사를 제거한 동사를 서로 비교하는, 자연어 요구사항을 이용한 메타모델

변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 8】

제1항에 있어서,

상기 원인-결과 그래프 생성부는,

상기 C3Tree 모델에 포함된 각 절(Clause)을 상기 원인 노드, 상기 결과 노드 및 이들의 연결 노드로 변환하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 9】

제1항에 있어서,

상기 결정 테이블 생성부는,

테이블 변환 엔진을 포함하고,

상기 테이블 변환 엔진은

상기 제1 메타모델 변환 기법을 통해 원인-결과 그래프 메타모델을 참고하여 상기 원인-결과 그래프 모델을 리딩하고, 결정 테이블 변환 룰을 기초로 결정 테이블 메타모델을 참조하여 상기 결정 테이블 모델을 생성하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 10】

제9항에 있어서,

상기 테스트 케이스 생성부는,

케이스 변환 엔진을 포함하고,

상기 케이스 변환 엔진은,

상기 제2 메타모델 변환 기법을 통해 상기 결정 테이블 메타모델을 참고하여 상기 결정 테이블 모델을 리딩하고, 테스트 케이스 변환 룰을 기초로 테스트 케이스 메타모델을 참조하여 상기 테스트 케이스 모델을 생성하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 11】

제10항에 있어서,

상기 테스트 케이스 모델은,

상기 테스트 케이스 모델을 실행하기 위해 상기 결정 테이블 모델에서 미리 실행되어야 하는 조합을 식별하여 생성되는 사전조건 정보를 포함하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 12】

제1항에 있어서,

상기 테스트 케이스 모델을 수신하여 이에 대한 테스트 요청 정보를 생성하고, 이를 티켓 프로그램에 전송 후 상기 테스트 요청 정보에 대한 응답 정보를 수신하는 테스트 실행부를 구비하는 테스트 장치를 더 포함하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 13】

제12항에 있어서,

상기 테스트 장치는,

상기 응답 정보를 기설정된 형식에 따라 상기 테스트 케이스 모델의 테스트 결과에 대한 테스트 결과 정보를 생성하는 테스트 결과부를 더 포함하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템.

【청구항 14】

문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과

정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계;

원인-결과 그래프 생성부를 통해 상기 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성하는 단계;

결정 테이블 생성부를 통해 상기 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환하는 단계; 및

테스트 케이스 생성부를 통해 상기 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 상기 입력값 및 결과값 중 적어도 어느 하나가 True인 조합으로 구성되는 테스트 케이스 모델을 생성하는 단계;를 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 15】

제14항에 있어서,

상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는,

상기 자연어 요구사항이 수동 문장인 경우, 이를 능동 문장으로 변화시키는 말뭉치 정규화 과정을 실시하는 단계를 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 16】

제15항에 있어서,

상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는,

상기 C3Tree 모델이 복수로 구성되는 경우, 상기 복수의 C3Tree 모델의 노드에 포함된 형태소의 유사도를 측정하여 형태소 유사도 기준값 이상인 경우 중복된 노드들을 병합시키는 단계를 더 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 17】

제16항에 있어서,

상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는,

상기 복수의 C3Tree 모델의 노드에 포함된 의미역을 서로 비교하여 의미가 중복되는 노드를 병합하는 단계를 더 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 18】

제17항에 있어서,

상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는,

부정 부사의 포함 여부를 판단하여 상기 부정 부사가 포함된 노드를 부정문으로, 상기 부정 부사가 포함되지 않은 노드를 긍정문으로 판단하여 상기 부정문과 상기 긍정문을 다른 의미로 판단하는 단계를 더 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 19】

제18항에 있어서,

상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는,

상기 복수의 C3Tree 모델의 노드에 포함된 상기 부정 부사의 개수를 분석하여 홀수 개인 경우 상기 부정문으로, 짝수 개인 경우 상기 긍정문으로 판단하는 단계를 더 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 20】

제19항에 있어서,

상기 문장 분석부를 통해 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 단계는,

상기 복수의 C3Tree 모델의 노드에 접미사가 포함된 동사가 존재하는 경우, 상기 접미사를 제거한 동사를 서로 비교하는 단계를 더 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 21】

제14항에 있어서,

상기 원인-결과 그래프 생성부를 통해 상기 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성하는 단계는,

상기 C3Tree 모델에 포함된 각 절(Clause)을 상기 원인 노드, 상기 결과 노드 및 이들의 연결 노드로 변환하는 단계를 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 22】

제14항에 있어서,

상기 결정 테이블 생성부는,

테이블 변환 엔진을 포함하고,

상기 결정 테이블 생성부를 통해 상기 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환하는 단계는,

상기 테이블 변환 엔진을 통해, 상기 제1 메타모델 변환 기법을 통해 원인-결과 그래프 메타모델을 참고하여 상기 원인-결과 그래프 모델을 리딩하고, 결정 테이블 변환 룰을 기초로 결정 테이블 메타모델을 참조하여 상기 결정 테이블 모델을 생성하는 단계를 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 23】

제22항에 있어서,

상기 테스트 케이스 생성부는,

케이스 변환 엔진을 포함하고,

상기 테스트 케이스 생성부를 통해 상기 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 상기 입력값 및 결과값 중 적어도 어느 하나가 True인 조합으로 구성되는 테스트 케이스 모델을 생성하는 단계는,

상기 케이스 변환 엔진을 통해 상기 제2 메타모델 변환 기법을 통해 상기 결정 테이블 메타모델을 참고하여 상기 결정 테이블 모델을 리딩하고, 테스트 케이스 변환 룰을 기초로 테스트 케이스 메타모델을 참조하여 상기 테스트 케이스 모델을 생성하는 단계를 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 24】

제23항에 있어서,

상기 테스트 케이스 모델은,

상기 테스트 케이스 모델을 실행하기 위해 상기 결정 테이블 모델에서 미리 실행되어야 하는 조합을 식별하여 생성되는 사전조건 정보를 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 25】

제14항에 있어서,

테스트 장치의 테스트 실행부를 통해 상기 테스트 케이스 모델을 수신하여 이에 대한 테스트 요청 정보를 생성하고, 이를 타겟 프로그램에 전송 후 상기 테스트 요청 정보에 대한 응답 정보를 수신하는 단계를 더 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생성 방법.

【청구항 26】

제25항에 있어서,

상기 테스트 장치의 테스트 결과부를 통해 상기 응답 정보를 기설정된 형식에 따라 상기 테스트 케이스 모델의 테스트 결과에 대한 테스트 결과 정보를 생성하는 단계를 더 포함하는, 테스트 케이스 시스템을 이용한 테스트 케이스 자동 생

성 방법.

【요약서】

【요약】

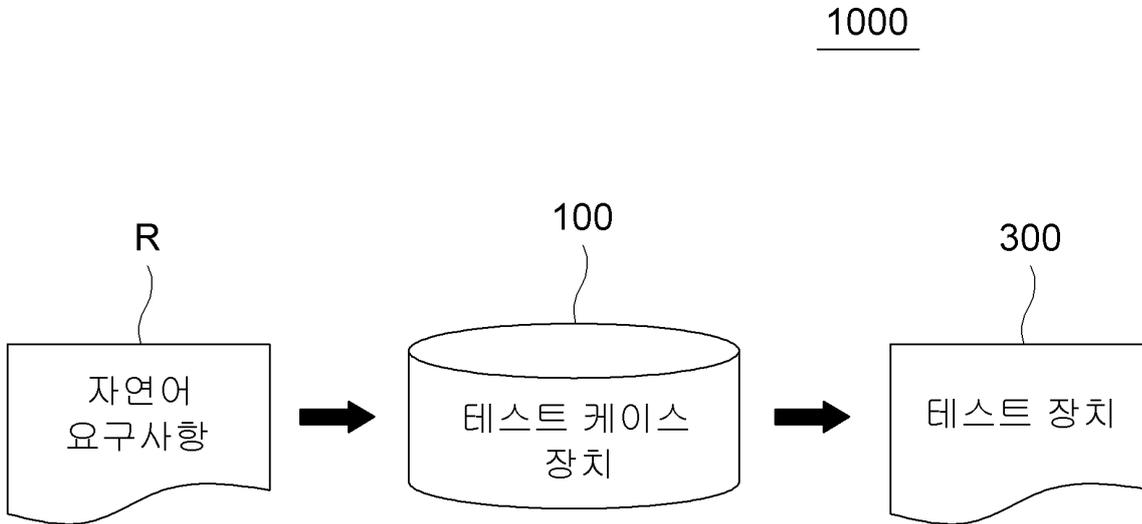
본 발명은 입력된 자연어 요구사항을 분석하여 문장의 절 분리 과정을 이진 트리 형태로 표현하는 C3Tree 모델을 생성하는 문장 분석부; 상기 C3Tree 모델을 기초로 원인-결과 그래프 모델을 생성하는 원인-결과 그래프 생성부; 상기 원인-결과 그래프 모델에 제1 메타모델 변환 기법을 적용하여 원인 노드와 결과 노드를 입력값 및 결과값의 True/False 조합으로 구성되는 결정 테이블 모델로 변환하는 결정 테이블 생성부; 및 상기 결정 테이블 모델에 제2 메타모델 변환 기법을 적용하여 상기 입력값 및 결과값 중 적어도 어느 하나가 True인 조합으로 구성되는 테스트 케이스 모델을 생성하는 테스트 케이스 생성부;를 포함하는, 자연어 요구사항을 이용한 메타모델 변환 방법을 기반으로 테스트 케이스를 자동으로 생성하는 테스트 케이스 시스템 및 이를 이용한 테스트 케이스 자동 생성 방법에 관한 것이다.

【대표도】

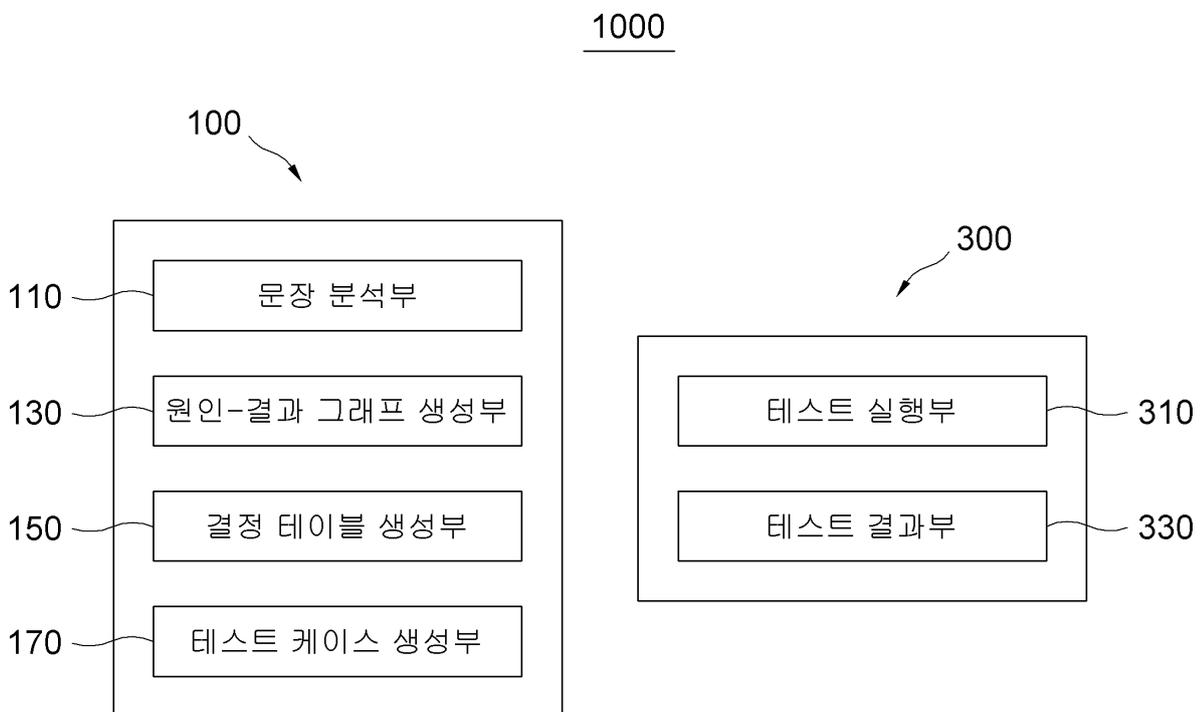
도 1

【도면】

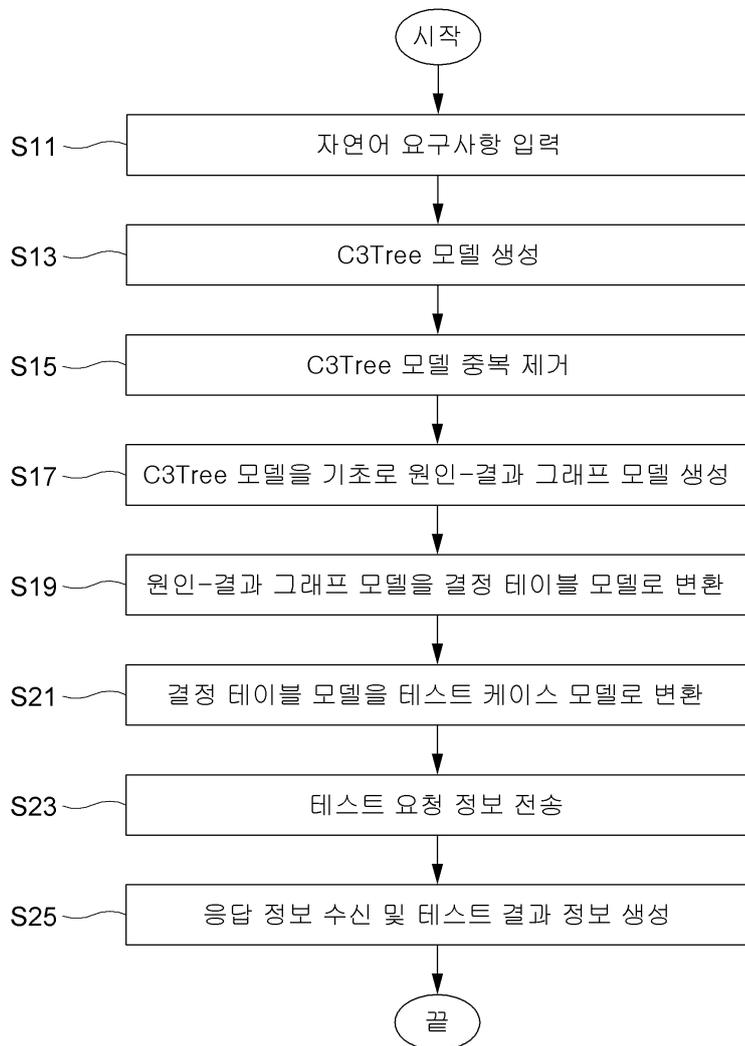
【도 1】



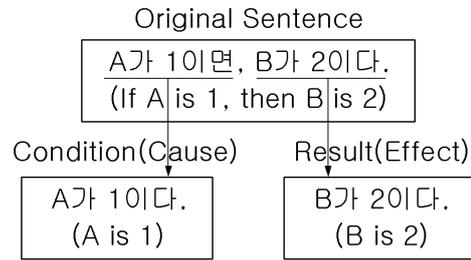
【도 2】



【도 3】



【도 4】



(a)

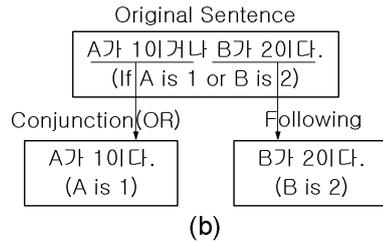
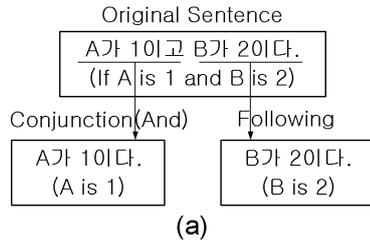
Condition Type	Subtype	Connective Ending Forms	Description
Opened condition	—	-면(Myun), -거든(Geodeun), ...	The veracity of conditions may change
Opened condition	Positive condition	-아(A)/어(Uh)/여야(Uya), ...	condition is true
	Negative condition	-던들(Deondeul), ...	condition is false

(b)

Type	Connective Ending Forms
Simple type	-면(Myun), -거든(Geodeun), -어야(Uya)
Complex type	-다면(Damyeon), -은들(Eundeul), -다가(Daga), -든지(Deunji), -려면(Lyeomyeon), -거드면(Geodeumyeon), -거들랑(Geodeullang), -대서야(Daeseoya), -고야(Goya), ...

(c)

【도 5】



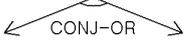
Conjunction Type	Connective Ending Forms	Logical Combination
Causality Relationship	‘-느라고(Nerago),’ Etc.	AND
Sequential relationship	‘-고(Go)(서(Seo)),’ Etc.	AND
Parallel relationship	‘-고(Go),’ ‘-(으(Eu))며(Myeo),’ ‘-(으(Eu))면서(Myeonseo),’ Etc.	AND
Contrast Relationship	‘-(으(Eu))나(Na),’ ‘-지만(Jiman),’ Etc.	OR

(c)

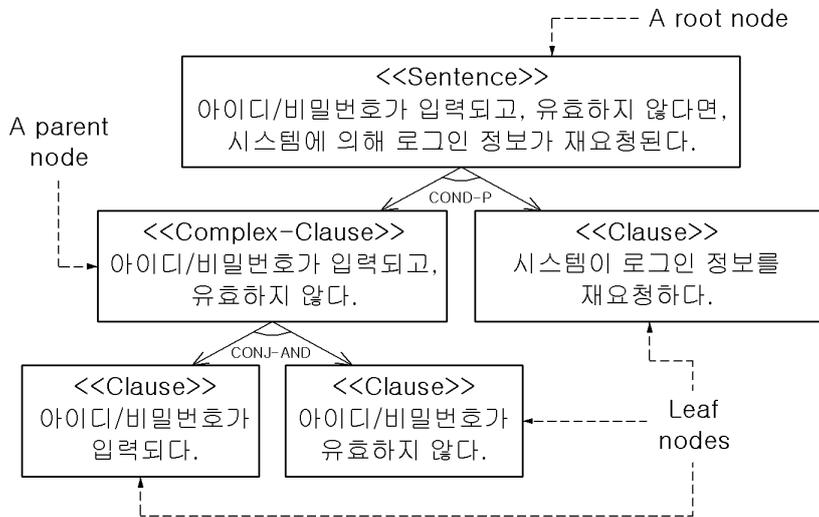
【도 7】

Node	Description
<<Sentence>> Korean Sentence	Root node. It contains the complex structure of the original sentence.
<<Complex-Clause>> Korean Sentence	Intermediate node. It is split from <<Sentence>>, and contains sentences that can be further split.
<<Clause>> Korean Sentence	Leaf node. It contains the most simplified sentences (short sentences).

(a)

Relationship	Description
	Child nodes are positive conditional relationships.
	Child nodes are negatively conditional relationships.
	Child nodes are conjunction(AND) relationship.
	Child nodes are conjunction(OR) relationship.

(b)



(c)

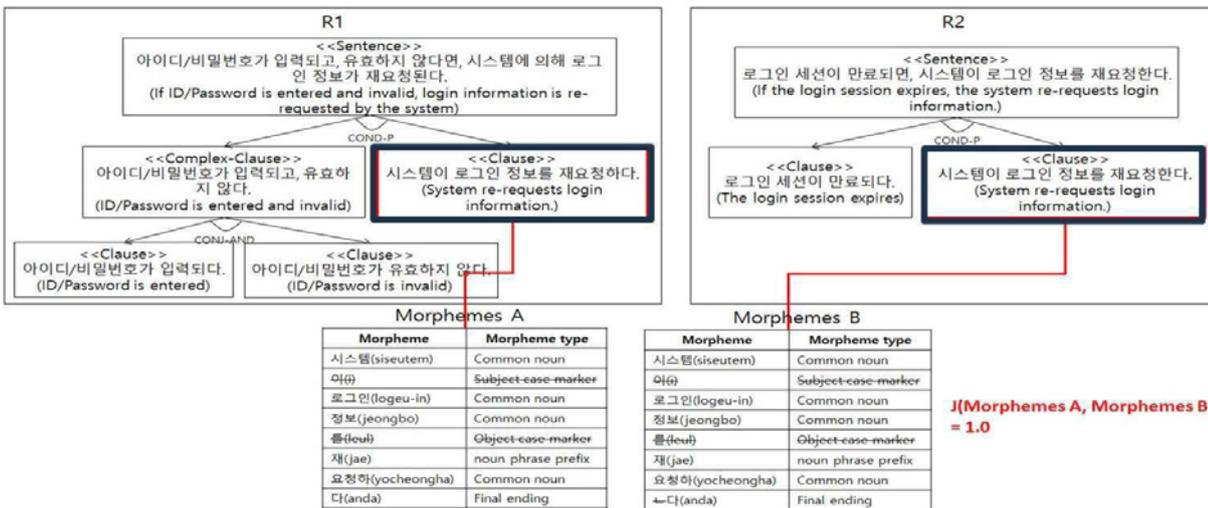
【도 8a】

$$J(A,B) = \frac{A \cap B}{A \cup B} = \frac{A \cap B}{|A| + |B| - |A \cap B|}$$

【도 8b】

Korean	로그인 세션이 만료되면, 시스템이 로그인 정보를 재요청한다.
English	When the login session expires, system re-requests login information.

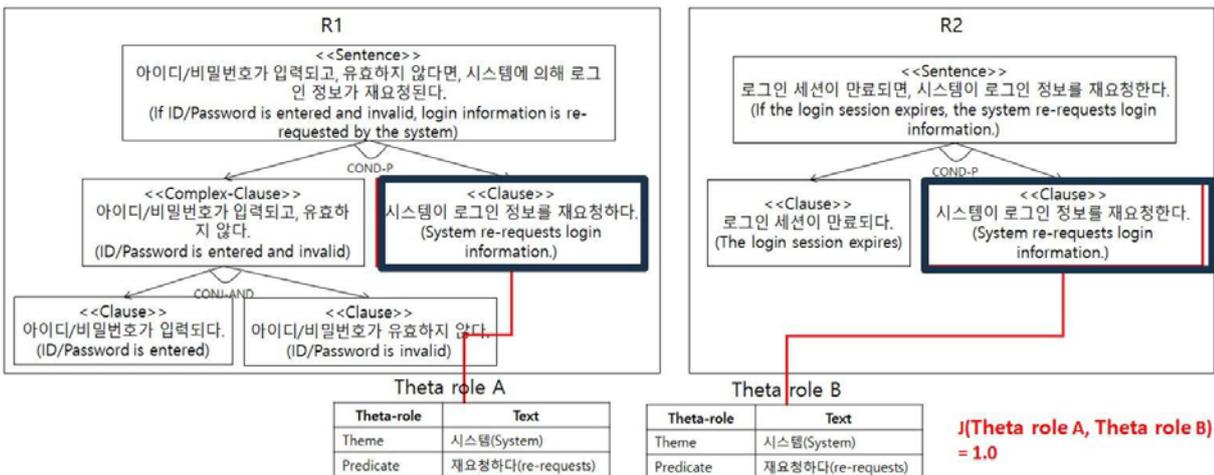
【도 8c】



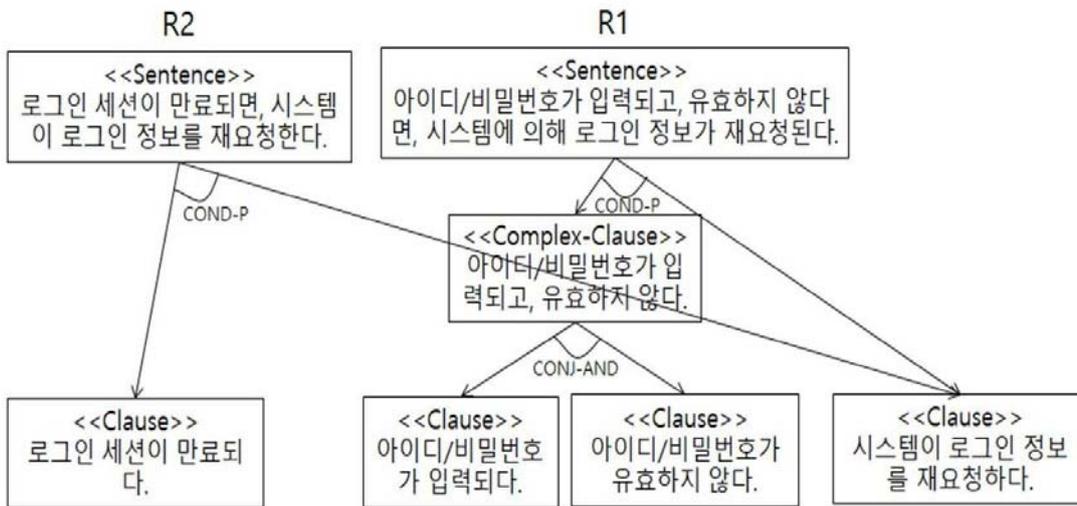
【도 9a】

Theta-role	Description
Agent	An object that causes an action with the intention expressed by the predicate.
Experiencer	The entity that recognizes an action or a state, not causing action with the intention.
Patient	The person or thing that undergoes the action.
Theme	An object that is the most central in the theta-role discussion. This is influenced by actions or processes, not controlling them.
Goal	The entity on activity that is directed
Source	The entity that starts a change when a predicate includes the identity of a person, a quality of a thing.
Instrument	The entity that indicates either a physical or abstract starting point when a verb includes a meaning related to moving or changing.

【도 9b】



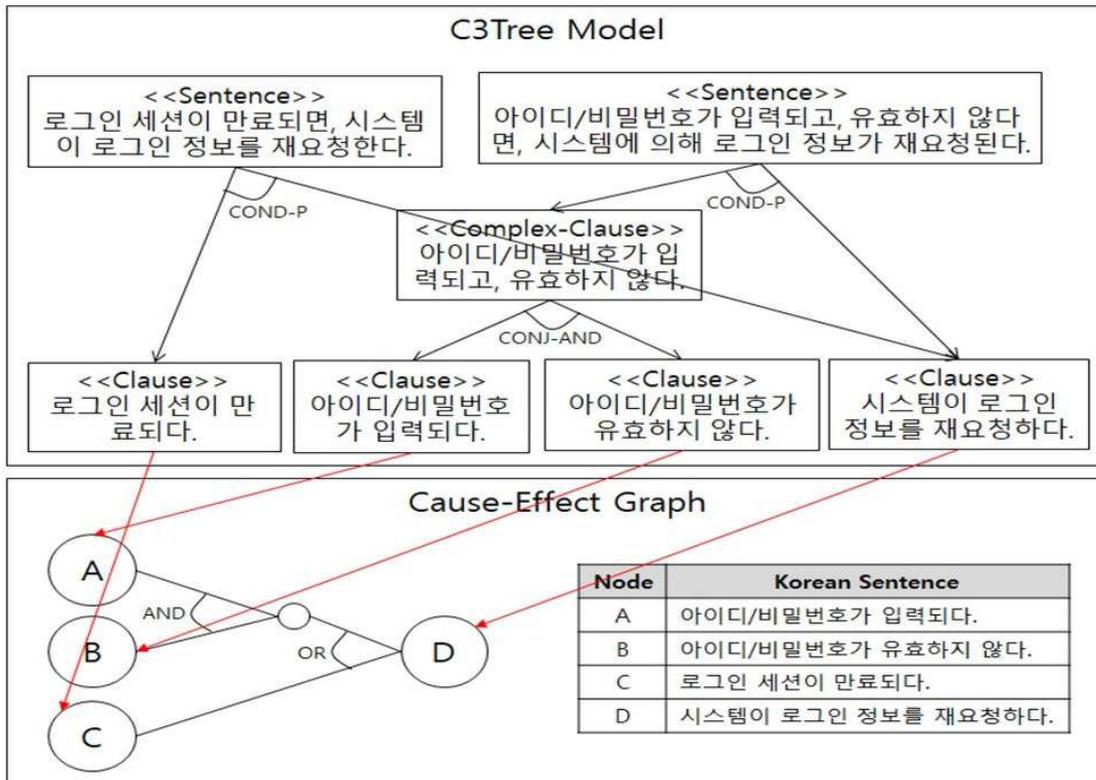
【도 9c】



【도 10a】

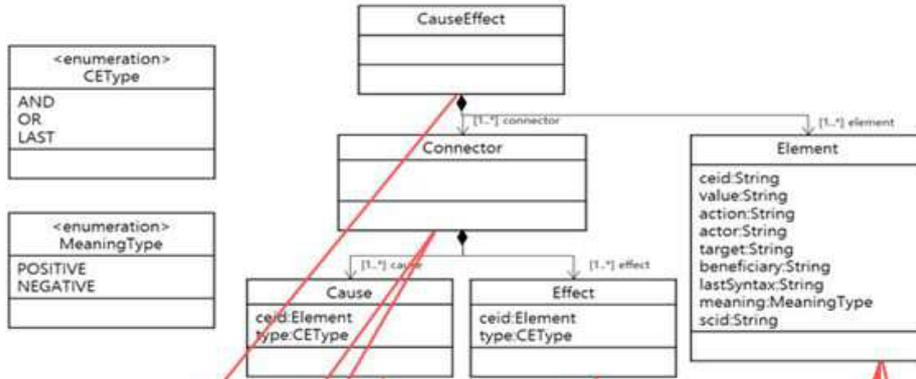
Case	C3Tree Model Notation	Cause-Effect Graph Notation
1	<<Clause>>	Node
2	CONJ-AND	AND
3	CONJ-OR	OR
4	The left child node of COND-P(COND-N)	Cause
5	The right child node of COND-P(COND-N)	Effect
6	COND-P	Identity
7	COND-N	NOT

【도 10b】



【도 11】

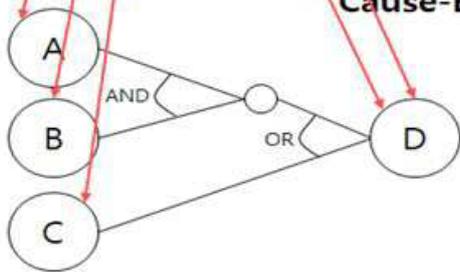
Cause-Effect Graph Metamodel



Cause-Effect Graph Model XMI Code Example

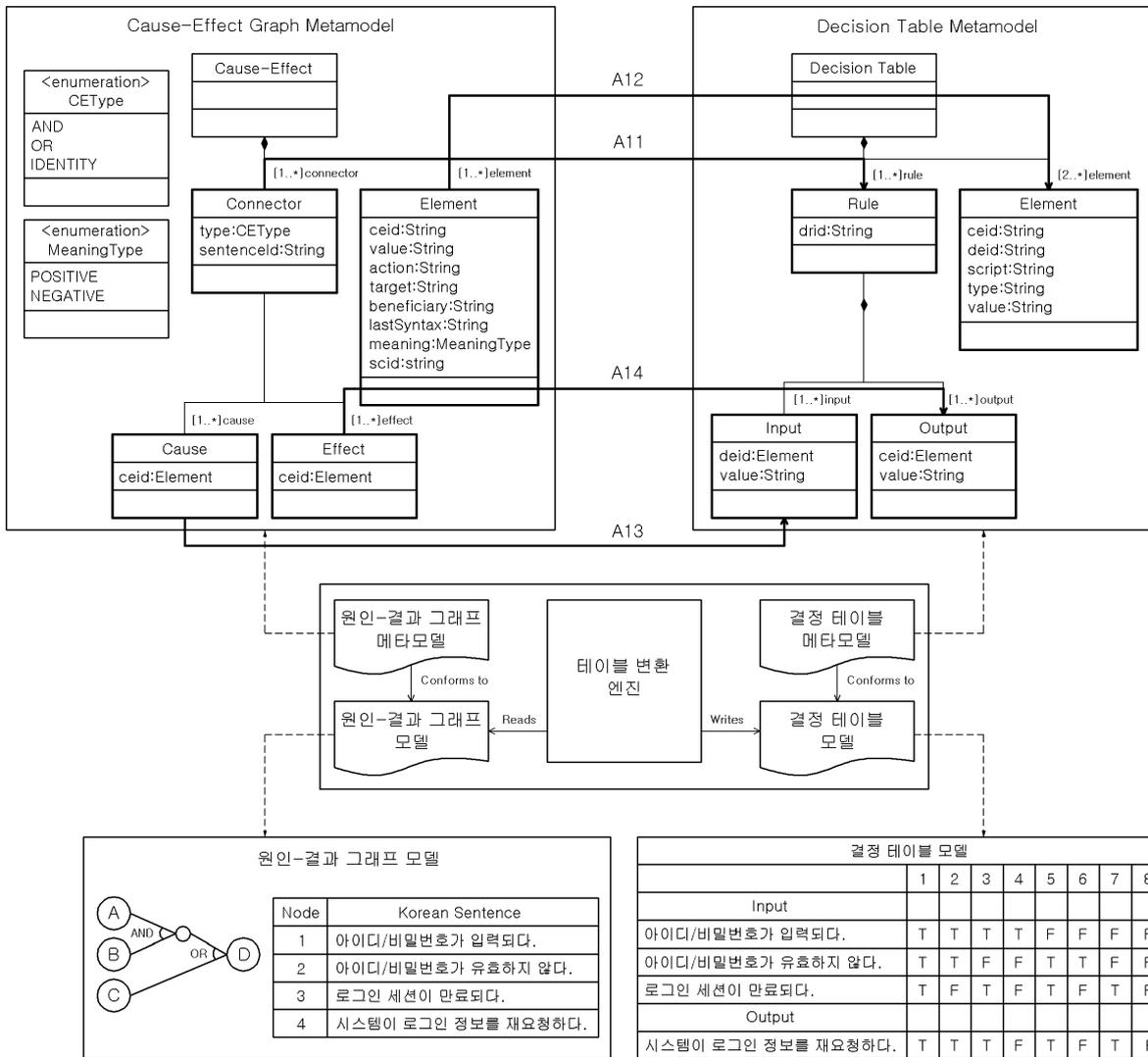
```
<?xml version="1.0" encoding=" UTF-8" ?>
<CauseEffectGraph>
  <element action="입력" actor="아이디/비밀번호" beneficiary="" ceid="0" lastsyntax=" 입력되다 "
  meaning="POSITIVE " scid=" 0 " target="" value="아이디/비밀번호가 입력되다."/>
  <element action=" 유효" actor=" 아이디/비밀번호" beneficiary=" " ceid=" 1 " lastsyntax=" 않다 "
  meaning="NEGATIVE" scid="1" target="" value="아이디/비밀번호가 유효하지 않다."/>
  <element action="만료" actor=" 로그인 세션" beneficiary="" ceid="2" lastsyntax="만료되다"
  meaning="POSITIVE" scid="2" target="" value="로그인 세션이 만료되다."/>
  <element action=" 재요청 " actor=" 시스템 " beneficiary=" " ceid=" 3 " lastsyntax=" 재요청하
  다 " meaning="" scid=" 4" target="로그인 정보" value="시스템이 로그인 정보를 재요청하다."/>
  <connector>
    <cause ceid="0" type="and"/>
    <cause ceid="1" type="last"/>
    <effect ceid="3" type="last"/>
  </connector>
  <connector>
    <cause ceid="2" type="last"/>
    <effect ceid="3" type="last"/>
  </connector>
</CauseEffectGraph>
```

Cause-Effect Graph



Node	Korean Sentence
A	아이디/비밀번호가 입력되다.
B	아이디/비밀번호가 유효하지 않다.
C	로그인 세션이 만료되다.
D	시스템이 로그인 정보를 재요청하다.

【도 12a】



【도 12b】

```

<?xml version="1.0" encoding="UTF-8"?>
<DecisionTable>
<element ceid="0" deid="0" script="" type="input" value="아이디/비밀번호가
입력된다."/>
<element ceid="1" deid="1" script="" type="input" value="아이디/비밀번호가
유효하지 않다."/>>
<element ceid="2" deid="2" script="" type="input" value="로그인 세션이
만료된다."/>
<element ceid="3" deid="3" script="" type="output" value="시스템이 로그인
정보를 재요청하다."/>
<rule drid="0">
<input deid="0" value="y"/>
<input deid="1" value="y"/>
<input deid="2" value="y"/>
<output deid="3" value="y"/>
</rule>
<rule drid="1">
<input deid="0" value="y"/>
<input deid="1" value="y"/>
<input deid="2" value="n"/>
<output deid="3" value="y"/>
</rule>
...
</DecisionTable>

```

【도 13a】

```

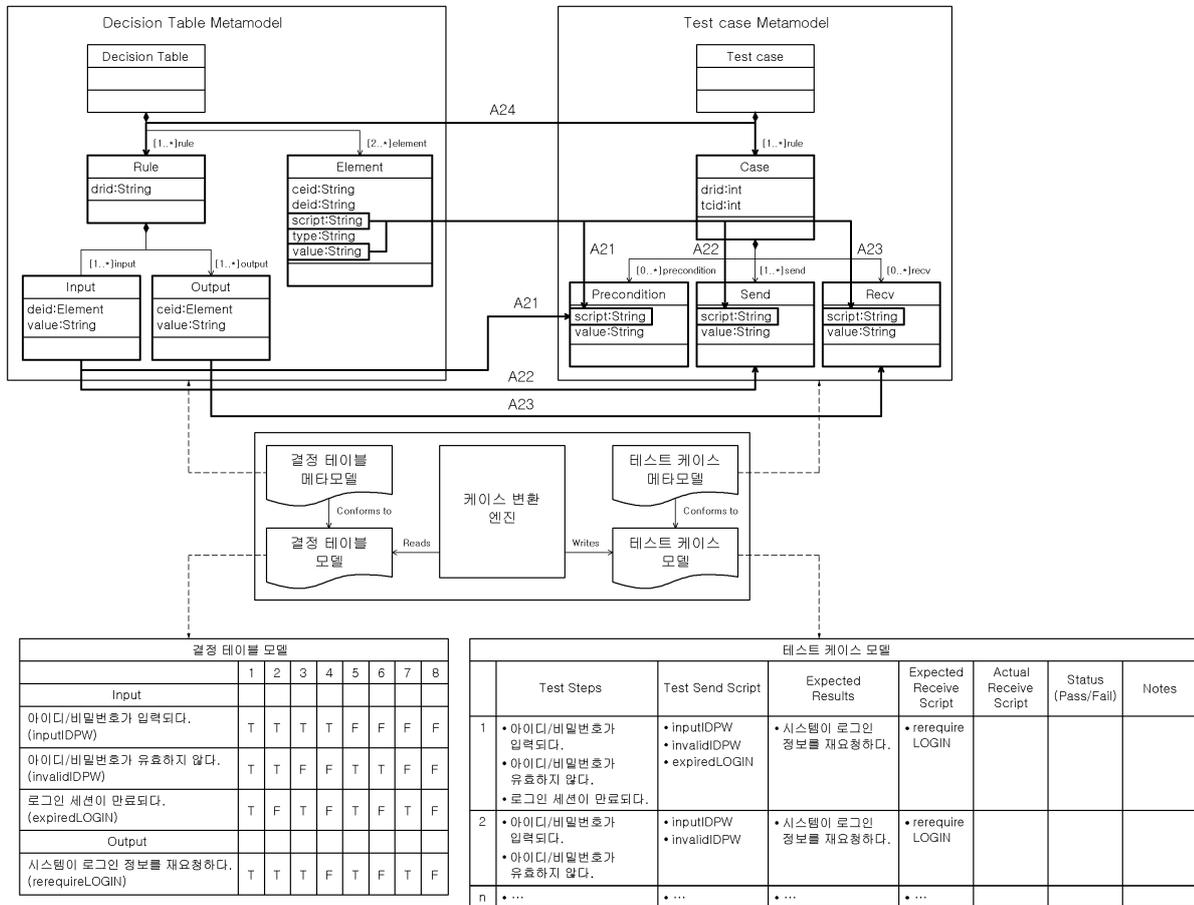
<?xml version="1.0" encoding="UTF-8"?>
<DecisionTable>
<element      ceid="0"      deid="0"      script="inputDPW"      type="input"
value="아이디/비밀번호가 입력된다."/>
<element      ceid="1"      deid="1"      script="invalidDPW"    type="input"
value="아이디/비밀번호가 유효하지 않다."/>
<element ceid="2" deid="2" script="expiredLOGIN" type="input" value="로그인
세션이 만료된다."/>
<element ceid="3" deid="3" script="rerequireLOGIN" type="output"
value="시스템이 로그인 정보를 재요청하다."/>
...
</DecisionTable>

```

【도 13b】

Decision Table								
	1	2	3	4	5	6	7	8
Input								
아이디/비밀번호가 입력된다. (inputDPW)	T	T	T	T	F	F	F	F
아이디/비밀번호가 유효하지 않다. (invalidDPW)	T	T	F	F	T	T	F	F
로그인 세션이 만료된다. (expiredLOGIN)	T	F	T	F	T	F	T	F
Output								
시스템이 로그인 정보를 재요청하다. (rerequireLOGIN)	T	T	T	F	T	F	T	F

【도 14a】

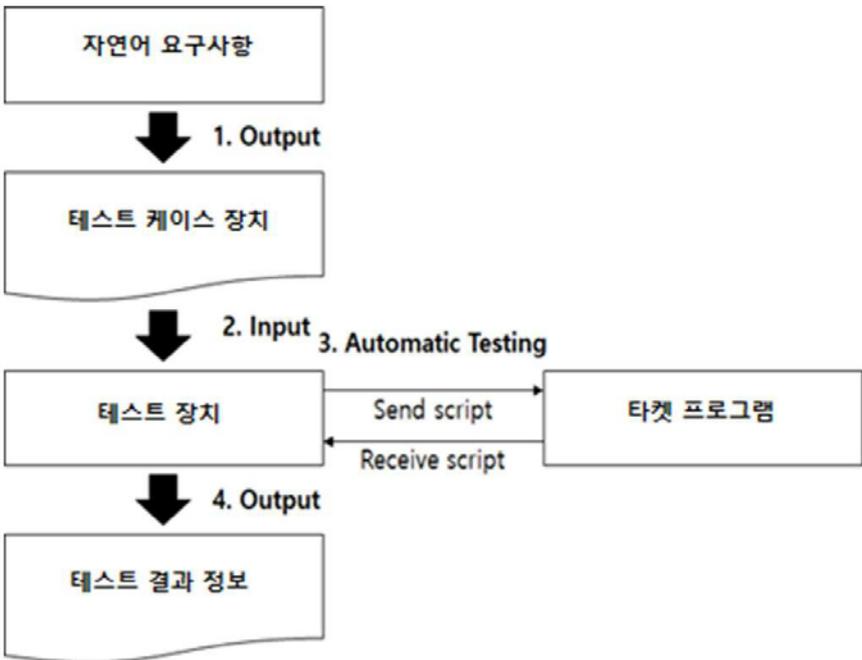


【도 14b】

```

<?xml version="1.0" encoding="UTF-8"?>
<Testcase>
<case drid="0" tcid="0">
<send script="inputIDPW" value="아이디/비밀번호가 입력된다."/>
<send script="invalidIDPW" value="아이디/비밀번호가 유효하지 않다."/>
<send script="expiredLOGIN" value="로그인 세션이 만료되었다."/>
<recv script="rerequireLOGIN" value="시스템이 로그인 정보를
재요청하다."/>
</case>
<case drid="1" tcid="1">
<send script="inputIDPW" value="아이디/비밀번호가 입력된다."/>
<send script="invalidIDPW" value="아이디/비밀번호가 유효하지 않다."/>
<recv script="rerequireLOGIN" value="시스템이 로그인 정보를
재요청하다."/>
</case>
...
</Testcase>
    
```

【도 15a】



【도 15b】

	Test Steps	Test Send Script	Expected Results	Expected Receive Script	Actual Receive Script	Status (Pass/Fail)	Notes
1	<ul style="list-style-type: none"> 아이디/비밀번호가 입력된다. 아이디/비밀번호가 유효하지 않다. 로그인 세션이 만료된다. 	<ul style="list-style-type: none"> inputIDPW invalidIDPW expiredLOGIN 	<ul style="list-style-type: none"> 시스템이 로그인 정보를 재요청한다. 	<ul style="list-style-type: none"> rerequireLOGIN 	<ul style="list-style-type: none"> rerequireLOGIN 	Pass	
2	<ul style="list-style-type: none"> 아이디/비밀번호가 입력된다. 아이디/비밀번호가 유효하지 않다. 	<ul style="list-style-type: none"> inputIDPW invalidIDPW 	<ul style="list-style-type: none"> 시스템이 로그인 정보를 재요청한다. 	<ul style="list-style-type: none"> rerequireLOGIN 	<ul style="list-style-type: none"> rerequireLOGIN 	Pass	
3	<ul style="list-style-type: none"> 아이디/비밀번호가 입력된다. 로그인 세션이 만료된다. 	<ul style="list-style-type: none"> inputIDPW expiredLOGIN 	<ul style="list-style-type: none"> 시스템이 로그인 정보를 재요청한다. 	<ul style="list-style-type: none"> rerequireLOGIN 	<ul style="list-style-type: none"> rerequireLOGIN 	Pass	
4	<ul style="list-style-type: none"> 아이디/비밀번호가 입력된다. 	<ul style="list-style-type: none"> inputIDPW 				Pass	
5	<ul style="list-style-type: none"> 아이디/비밀번호가 유효하지 않다. 로그인 세션이 만료된다. 	<ul style="list-style-type: none"> invalidIDPW expiredLOGIN 	<ul style="list-style-type: none"> 시스템이 로그인 정보를 재요청한다. 	<ul style="list-style-type: none"> rerequireLOGIN 	<ul style="list-style-type: none"> rerequireLOGIN 	Pass	
6	<ul style="list-style-type: none"> 아이디/비밀번호가 유효하지 않다. 	<ul style="list-style-type: none"> invalidIDPW 				Pass	
7	<ul style="list-style-type: none"> 로그인 세션이 만료된다. 	<ul style="list-style-type: none"> expiredLOGIN 	<ul style="list-style-type: none"> 시스템이 로그인 정보를 재요청한다. 	<ul style="list-style-type: none"> rerequireLOGIN 	<ul style="list-style-type: none"> rerequireLOGIN 	Pass	