제19권 제2호
ISSN 2005-0011

秋

Fall Conference 2012

제 38회 한국정보처리학회

추계학술발표대회

논문집(하)

일 자 : 2012년 11월 22일(목)~23일(금)

장 소 : **제주대학교** 아라캠퍼스

주 최 : 사단법인 **한국정보처리학회**

주 관 : 제주대학교 JEJU NATIONAL UNIVERSITY

후 원 : nipa 정보통신산업진흥원 National IT Industry Promotion Agency

협 찬 : 롯데정보통신, 삼성SDS, SK텔레콤, 굿모닝아이텍, 굿센테크날러지, 닉스테크, 비트컴퓨터, 영우디지탈, 유비벨록스, 이니텍, 제이컴정보, 콤텍시스템, 테크그룹, 티맥스소프트, 한국IT감리컨설팅, 한글과컴퓨터, 효성인포메이션시스템, NHN, KCC정보통신

사단법인 한국정보처리학회
KIPS Korea Information Processing Society

소프트웨어공학

407. AXIS Tool을 활용한 신호처리개발에 관한 연구 KIPS_C2012J_0033
김도훈*, 정기현(아주대학교) • 1415

408. 클라우드 컴퓨팅 환경에서 사용자 서비스의 시나리오 기반 테스트 케이스 생성 KIPS_C2012J_0045
김종필*, 홍장의(충북대학교) • 1418

409. 공공 SW사업의 성공적 수행을 위한 관계법령의 적합성 연구 KIPS_C2012J_0064
배성우*, 이석주(고려대학교) • 1421

410. 프로젝트 성공 촉진을 위한 프로젝트관리조직(PMO)에 관한 연구 KIPS_C2012J_0069
김태우*, 이석주(고려대학교) • 1425

411. 소프트웨어 규모산정을 위한 기능점수 측정 개선사례 KIPS_C2012J_0070
박종모*, 김승권(정보통신산업진흥원) • 1429

412. 프로젝트 성공을 위한 범위기술서, WBS관련 PI, KPI 지표에 관한 연구 KIPS_C2012J_0077
심미진*, 이석주(고려대학교) • 1433

413. 소프트웨어 저작권 확보를 위한 공개소프트웨어 라이선스 설정방안에 관한 연구 KIPS_C2012J_0107
박준석*, 김수홍(상명대학교) • 1437

414. SW공학수준, DICE수준과 SW 개발 프로젝트 납기성과 KIPS_C2012J_0118
김승권*, 이재덕(정보통신산업진흥원) • 1441

415. 오픈소스 형상도구와 결함추적도구를 이용한 개발 프로젝트 관리에 관한 연구 KIPS_C2012J_0126
이지현*, 박영식(한국전자통신연구원) • 1445

416. 파노라마 가상현실을 기반으로 하는 호서대학교 사이버 박물관의 설계 및 구현 KIPS_C2012J_0156
홍성수, 김창기*, 이르판 칸(호서대학교) • 1448

417. 지능형 교통 RoboCAR 운행제어 검증을 위한 Test Case 추출 및 적용 연구 KIPS_C2012J_0164
장우성*, 박찬민, 이철희(한백전자), 김영철(홍익대학교) • 1452

418. 온라인 게임의 사설서버 피해와 방어 KIPS_C2012J_0166
배정일*, 오상석, 민성기(고려대학교) • 1456

419. 국방무기체계 모델 개발을 위한 SysML 적용 사례 연구 KIPS_C2012J_0190
장재덕*, 최상택, 정윤호, 최상욱(LIG넥스원㈜), 최광묵(MDS 테크놀로지㈜) • 1460

420. MobileMapGen : 모바일 맵 응용 생성기 KIPS_C2012J_0234
민경윤*, 박원진, 음두헌(덕성여자대학교) • 1464

421. A study on Web-based Video Panoramic Virtual Reality for Hose Cyber Shell Museum KIPS_C2012J_0252
Sung-Soo Hong*, Irfan khan, Chang-ki Kim(Hoseo University) • 1468

422. 전력 R&D SW 엔지니어링 플랫폼 개발 KIPS_C2012J_0266
주용재*, 송재주, 최민희(한국전력공사) • 1472

423. 전력 R&D SW 표준프레임워크 구조 설계 KIPS_C2012J_0268
송재주, 주용재*(한국전력공사) • 1475

424. 항공관제에서 비행자료 보호 관리 시스템에 관한 연구 KIPS_C2012J_0270
이덕규*, 한종욱(한국전자통신연구원) • 1479

425. 국방무기체계 개발을 위한 요구사항 개발/관리 프로세스 정립 및 도구 개선 사례 연구 KIPS_C2012J_0272
최상욱*, 최상택, 정윤호, 장재덕(LIG넥스원) • 1482

우수논문 426. 컨텍스트 인지 모바일 컴퓨팅을 위한 범용 추론 시스템의 정형 모델 KIPS_C2012J_0290
김문권*, 김수동(숭실대학교) • 1486

지능형 교통 RoboCAR 운행제어 검증을 위한 Test Case 추출 및 적용 연구

장우성*, 박찬민*, 이철희*, 김영철**
*한백전자
**홍익대학교 소프트웨어공학연구실
{wsjang, eacs, chlee}@hanback.co.kr*, bob@selab.hongik.ac.kr**

A Study on Test Case Extraction And Application for Intelligent Transport RoboCAR Drive Control Verification

Woo-Sung Jang*, Chan-Min Park*, Cheul-Hee Lee*, R.Young-Chul Kim**
*Hanback Electronics
**Dept. of Software Engineering, Hong-ik University

요 약

본 논문은 출시된 지능형 교통 기반으로 설계된 RoboCAR 운행제어 개발과 시험을 하고자 한다. 이를 위해 시스템 설계를 통해,Test Case 추출과 실제로 적용하여 구현된 소프트웨어를 시험에 목적을 둔다. 이 절차는 Use-Case Diagram 설계, Decision Factor 추출, 이 기반으로 Cause-Effect Diagram 을 생성한다. Cause-Effect Diagram 을 통해 Decision Table 을 생성한다. 최종적으로 Decision Table 을 기반으로 Test Case 를 추출한다. 추출된 Test Case 를 적용하여 시스템을 테스트 하였고, 설계와 구현이 동일하게 되었음을 검증하였다.

1. 서론

소프트웨어 개발은 설계 단계에서부터 유지보수 단계에 이르기까지 관리가 필요하다. 1970 년 초기에는 소프트웨어 개발이란 관점에서 주력했다. 하지만 지금은 유지보수에 더 큰 관심을 가지고 있다. Boehm 에 의하면 유지보수 비용이 컴퓨터 시스템의 모든 비용 중 큰 비중을 차지하고 있고, 앞으로 계속적인 유지보수 비용의 증가로 인하여 개발보다는 유지보수 비용에 많은 돈이 투자될 것이다[1]. 이 유지보수 비용을 줄이기 위해서는 체계적인 소프트웨어 테스트가 필요하다. 소프트웨어 테스트는 소프트웨어 개발 프로세스의 각 단계에서 수행 되어지는 작업의 품질을 측정하고 평가하기 위한 행위로서 소프트웨어가 계획된 대로 수행되는지를 확인하여 소프트웨어의 품질과 신뢰성 그리고 유지보수성을 향상시키는데 목적이 있다[2].

체계적인 소프트웨어 테스트는 정확한 요구사항으로부터 Test Case 를 추출하여 개발 후 시스템 검증 테스트가 필요하다. 기존 Test Case 발생 방법으로는 여러 가지가 있으나, 대부분 구조적 개발 방법 상에서의 연구가 많다. 특히 객체지향 및 Use-Case 개발환경에서 State Machine 을 통하여 Test Case 발생방법, Use-Case Diagram 을 통한 Test Case 발생 방법 등이 있다[3].

본 논문은 객체지향으로 설계된 지능형 교통 정보 운행 시스템에서 Test Case 를 발생하여 시스템의 유지보수 비용을 줄이는데 목적을 둔다. Use-Case Diagram 으로 설계된 시스템을 Cause-Effect Diagram 을 통해 Test Case 를 발생시킨다. Cause-Effect Diagram 을 통한 Test Case 발생은 Gary E.Mogyorodi 가 제안한 검증된 방법이다. 이 방법은 최소의 Test Case 로 100% 기능적인 요구사항 커버리지를 만족 시킬 수 있는 방법이다[4]. 하지만 구조적 설계 기반이기 때문에 객체지향 시스템에는 적합하지 않다. 이 문제점은 Use-Case Diagram 에서 Decision Factor 를 추출하여 Cause-Effect Diagram 에 적용하는 방법으로 해결한다[2].

2. 관련연구

2.1. Intelligent Transport System

지능형 교통 시스템(Intelligent Transport System, ITS)이란, 교통• 전자• 통신• 제어 등 첨단기술을 도로• 차량• 화물 등 교통체계의 구성요소에 적용하여 실시간 교통정보를 수집• 관리• 제공함으로써, 교통시설의 이용효율을 극대화하고, 교통이용편의와 교통안전을 제고하며, 에너지 절감 등 환경친화적 교통체계를 구현하는 21 세기형 교통체계를 의미한다[5].

ITS 서비스는 그 목적에 따라 교통관리체계, 교통정보체계, 그리고 대중교통체계, 화물운송체계, 차세대 차량 및 도로체계로 구분할 수 있다[6].

교통관리체계는 도로 및 차량에 교통 상황, 차량 정보를 감지할 수 있는 장치를 설치하여 수집한 교통정보를 토대로 교통신호운영을 최적화하고, 교통흐름을 유연하게 하는 교통체계이다.

교통정보체계는 운전자나 여행자가 원하는 각종 적시적소에 제공하여 목적지까지의 최단경로안내를 비롯한 각종 여행편의를 제공하고 교통의 분산을 유도하는 체계이다.

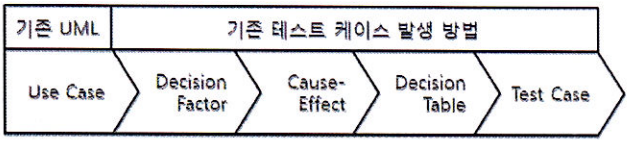
대중교통체계는 대중교통 노선정보 및 운행정보를 수집하여 운영자 및 이용자에게 제공하여 운영효율을 높이고, 이용자의 편의를 증진시켜 대중교통 이용의 활성화를 도모하는 체계이다.

화물운송체계는 화물 차량의 정보를 자동으로 파악하여 화물차 운행을 최적화하고 관리를 효율화시키는 체계이다.

차세대 차량 및 도로체계는 차량에 고성능 센서와 자동제어장치를 부착하여 전방, 측방추돌을 예방하고, 차선이탈 방지와 차량운행거리를 좁혀 도로소통능력을 배가시키는 교통체계이다.

2.2. Test Case 추출 절차

아래의 그림 1 은 본 논문에 적용된 Test Case 추출 절차로 Use-Case 개발기반으로, Use Case Diagram 으로 설계된 시스템을 Gary E. Mogyorodi 의 Cause-Effect Graphing 기술을 통해 Test Case 를 추출한다. Cause-Effect Graphing 기술은 최소의 Test Case 로 100%의 기능적인 요구사항 커버리지를 만족시킬 수 있는 검증된 방법이다[3]. 하지만 구조적 설계를 기반으로 하기 때문에 객체지향 개념인 Use-Case Diagram 을 적용하기 위해서는 Use-Case 의 결정요소를 추출하는 과정이 필요하다.



(그림 1) Test Case 추출 프로세스

첫 번째 단계는 UML 언어 중 Use-Case Diagram 을 그리고, Description 을 정의한다.

두 번째 단계는 Use-Case Description 에서 Decision Factor 를 추출한다. 추출된 Decision Factor 의 종류는 Input, Condition, Output 이다. Input 은 Output 이 나오기 위한 입력값이다. Condition 은 Output 이 나오기 위한 Input 값의 조건이다. Output 은 Input 에 의해 출력되는 결과값이다.

세 번째 단계는 추출된 Decision Factor 를 기반으로 Cause-Effect Diagram 을 생성한다.

네 번째 단계는 Cause-Effect Diagram 을 기반으로 Decision Table 을 생성한다. Decision Table 은 논리적인 조건을 포함하고 있는 시스템의 요구사항과 내부적인 시스템의 설계를 문서화 하는 방법이다. 이때 테이블이 아닌 트리 형식인 그래픽도 가능하다. 하지만 그래픽 형식의 단점은 nodes(causes and effects)나 branches 는 잘못 설정이나, missing 된 노드와 branch 를 찾을 수 없다. 그래서 결정 트리가 아닌 결정 테이블을 사용한다[3].

다섯 번째 단계는 Decision Table 을 기반으로 Test Case 를 생성한다.

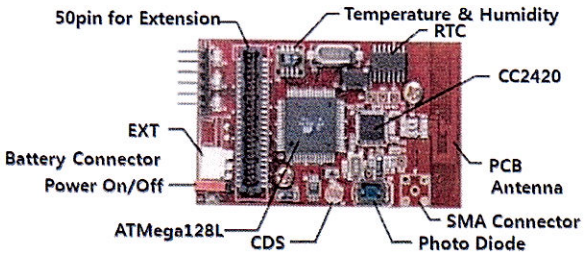
3. ITS 기반 RoboCAR 운행제어 시스템

3.1. 시스템 구성요소

RoboCAR Control System 은 차량의 움직임을 원격으로 제어하고, 교통 정보를 실시간으로 감시하는 모의 환경을 구축한 교육용 시스템이다. 16 개의 신호등, 4 개의 사거리, 1 개의 버스정류장이 설치된 모의 도로 환경에서 HBE-RoboCAR-Vision 을 이용해 모의 주행이 가능하다. 시스템을 구성하는 장비는 HBE-RoboCAR-Vision, HBE-ZigbeX, PLC 이다.

HBE-RoboCAR-Vision 은 카메라가 장착된 4 륜 차량형 로봇이다. 적외선 센서, 기울기 센서, 초음파 센서가 내장되어 있다. 적외선 센서 기반의 라인트레이서 기능을 사용하여 도로를 주행한다.

HBE-ZigbeX 는 유비쿼터스 센서 네트워크(Ubiquitous Sensor Network) 환경에서 다양한 실습이 가능하도록 제작된 장비이다. 다양한 센서가 내장되며, 주변 센서 노드들과 협업하여 Ad-hoc 통신 기법으로 데이터를 전송한다.[7]. HBE-ZigbeX 는 각 신호등 rhk 버스정류장, PLC, RoboCAR-Vision 에 장착되어 있고, 외형은 아래의 그림 2 와 같다.

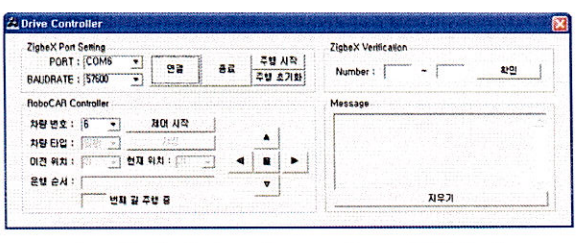


(그림 2) HBE-ZigbeX

PLC 는 신호등의 신호를 정해진 순서에 따라 제어한다. 모의 환경 내에 총 4 개의 PLC 가 설치되어 있으며, 한 개의 PLC 는 4 개의 신호등을 제어한다.

3.2. Drive Controller

Drive Controller 는 RoboCAR 운행제어 시스템의 소프트웨어이다. 신호등, PLC 와 통신을 하여 현재 도로 정보를 분석하고, 분석한 정보를 토대로 HBE-RoboCAR-Vision 을 제어한다. 실행 화면은 아래의 그림 3 과 같다.



(그림 3) Drive Controller 실행화면

Drive Controller 는 크게 ZigbeX Port Setting, RoboCAR Controller, ZigbeX Verification, Message 의 네 가지 기능으로 나뉜다.

ZigbeX Port Setting 은 ZigbeX 와 연결되는 COM 포트를 설정하고, RoboCAR-Vision 의 주행을 시작하거나, 주행 정보를 초기화 한다.

RoboCAR Controller 는 주행 중인 RoboCAR-Vision 중 한 대를 선택하여 제어한다. 현재 차량의 주행 순서를 정의할 수 있고, 수동 컨트롤도 가능하다.

ZigbeX Verification 은 ZigbeX 가 정상적으로 동작하는지 확인한다. 확인 메시지는 Message 기능에서 확인할 수 있다.

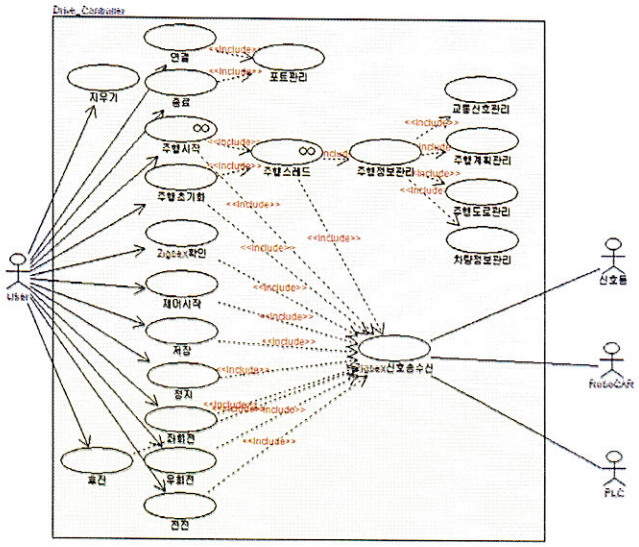
Message 는 각 장비에 연결된 ZigbeX 들이 PC 에게 보내는 메시지를 출력한다.

4. RoboCAR 운행 제어 시스템 Test Case 추출

4.1. Use-Case Diagram 설계

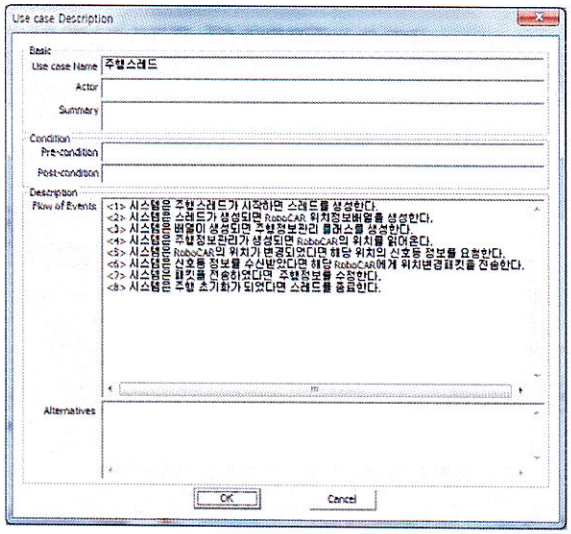
Drive Controller 를 Use-Case Diagram 으로 표현한 결과는 아래의 그림 4 와 같다. 그림 4 의 Diagram 은 홍익대학교의 HiMEM 을 이용해 표현하였다. HiMEM 은 확장된 UML Diagram 을 쉽게 그리고, Test Case 로 자동 변환하는 기능을 지원하는 프로그램이다.

Drive Controller 의 Use-Case 는 연결, 종료, 주행시작, 주행초기화, ZigbeX 확인, 제어시작, 저장, 정지, 좌회전, 우회전, 전진, 후진, 지우기, 주행스레드, 주행정보관리, 교통신호관리, 주행계획관리, 주행도로관리, 차량정보관리, ZigbeX 신호 송수신으로 구성된다. User 는 프로그램의 버튼 기능에 해당하는 Use-Case 에 접근 가능하고, 신호등, RoboCAR, PLC 는 ZigbeX 신호 송수신 Use-Case 에 접근 가능하다.



(그림 4) Use-Case Diagram

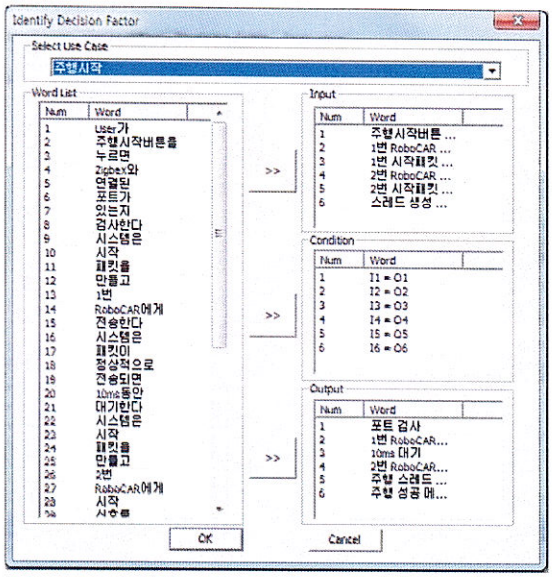
각 Use-Case 는 Description 을 가지고 있다. Use-Case Description 을 설정한 결과는 아래의 그림 5 와 같다. 본 논문은 정의된 여러 개의 Use-Case 중 주행 스레드 Use-Case 만 설명한다.



(그림 5) Use-Case Description

4.2. Decision Factor 추출

Use-Case 에서 Decision Factor 를 추출하기 위해서는 정의된 Description 에서 Input, Condition, Output 을 추출해내야 한다. 현재 HiMEM 은 완벽한 자동 추출이 불가능하고, 문장을 띄어쓰기 기준으로 하나의 요소로 분리해준다. 사용자는 이 요소들을 수동으로 분류하여 Input, Condition, Output 결정한다. HiMEM 이 분류한 요소들의 리스트는 아래 그림 6 의 좌측 리스트와 같이 분류된다.



(그림 6) Decision Factor 추출

Decision Factor 를 추출한 결과는 아래의 그림 7 과 같다. I1~I6 은 input 을 의미한다. O1~O5 는 output 을 의미한다. C1~C6 은 condition 을 의미한다.

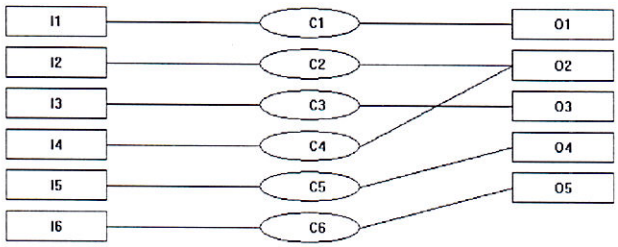
List Num	Value	Type
I1	주행시작버튼 클릭	input
I2	1번 RoboCAR 시작 패킷 생성	input
I3	1번 시작패킷 전송 완료	input
I4	2번 RoboCAR 시작 패킷 생성	input
I5	2번 시작패킷 전송 완료	input
I6	스레드 생성 완료	input
O1	포트 접사	output
O2	RoboCAR에게 전송	output
O3	10Ms 대기	output
O4	주행 스레드 생성	output
O5	주행 성공 메시지 출력	output
C1	I1 = O1	condition
C2	I2 = O2	condition
C3	I3 = O3	condition
C4	I4 = O2	condition
C5	I5 = O4	condition
C6	I6 = O5	condition

(그림 7) 추출된 Decision Factor

4.3. Cause-Effect Diagram 생성

Decision Factor 가 추출되었다면, 이를 기반으로 Cause-Effect Diagram 을 그릴 수 있다. Cause-Effect Diagram 은 HiMEM 에서 자동으로 생성된다.

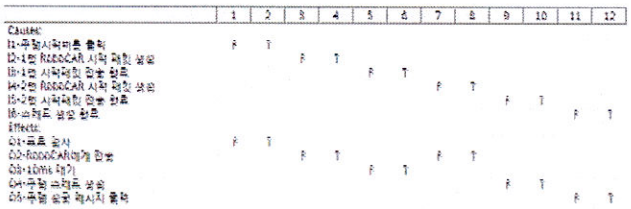
그림 7 을 기반으로 생성된 Cause-Effect Diagram 은 아래의 그림 8 과 같다. Cause-Effect Diagram 의 I, C, O 는 Decision Factor 의 I, C, O 와 동일한 의미이다.



(그림 8) Cause-Effect Diagram

4.4. Decision Table 생성

Cause-Effect Diagram 을 생성하였다면, 이를 기반으로 Decision Table 을 생성할 수 있다. Decision Table 또한 HiMEM 에서 자동 생성된다. Decision Table 은 아래의 그림 9 와 같다. 입력의 Condition 에 따라 출력의 결과가 달라지는 것을 테이블로 나타낸다.



(그림 9) Decision Table

4.5. Test Case 추출

Decision Table 을 기반으로 Test Case 를 생성한다. Test Case 는 아래의 그림 10 과 같다. Test Case 또한 Decision Table 이 존재한다면 자동으로 생성된다. Decision Table 에서 출력된 모든 경우의 수를 Test Case 로 나타낸다. 생성된 Test Case 는 주행 스레드 Use-Case 에 대한 Test Case 이다. 이 외에도 다른 Use-Case 의 Test Case 를 생성하여 Drive Control 프로그램의 테스트를 진행할 수 있다.

No	Post Condition	Test Condition	Expectation Result
TC 1	I1-주행시작버튼 클릭=F	N/A	O1-포트 접사=F
TC 2	I1-주행시작버튼 클릭=T	N/A	O1-포트 접사=T
TC 3	I2-1번 RoboCAR 시작 패킷 생성=F	N/A	O2-RoboCAR에게 전송=F
TC 4	I2-1번 RoboCAR 시작 패킷 생성=T	N/A	O2-RoboCAR에게 전송=T
TC 5	I3-1번 시작패킷 전송 완료=F	N/A	O3-10ms 대기=F
TC 6	I3-1번 시작패킷 전송 완료=T	N/A	O3-10ms 대기=T
TC 7	I4-2번 RoboCAR 시작 패킷 생성=F	N/A	O2-RoboCAR에게 전송=F
TC 8	I4-2번 RoboCAR 시작 패킷 생성=T	N/A	O2-RoboCAR에게 전송=T
TC 9	I5-2번 시작패킷 전송 완료=F	N/A	O4-주행 스레드 생성=F
TC 10	I5-2번 시작패킷 전송 완료=T	N/A	O4-주행 스레드 생성=T
TC 11	I6-스레드 생성 완료=F	N/A	O5-주행 성공 메시지 출력=F
TC 12	I6-스레드 생성 완료=T	N/A	O5-주행 성공 메시지 출력=T

(그림 10) 생성된 Test Case

5. 결론

본 논문에서는 ITS 기반의 RoboCAR 운행 제어 시스템 중 Drive Controller 프로그램에 대한 Test Case 를 추출하여 프로그램 기능을 검증하였다. 첫 번째로 Use-Case Diagram 에서 Decision Factor 를 추출하였다. 두 번째로 Decision Factor 를 Cause-Effect Diagram 으로 변환하였다. 세 번째로 Cause-Effect Diagram 을 Decision Table 로 변환하였다. 최종적으로 Decision Table 에서 Test Case 를 추출하였다. 추출된 Test Case 를 적용하여 프로그램이 실제 설계대로 구현이 되었는지 검증하였다. 이 방법을 통해 객체 지향으로 설계된 시스템의 안정성 테스트가 가능하였다.

향후 연구는 소프트웨어와 로봇의 펌웨어를 함께 테스트 할 수 있는 Test Case 추출 방법 연구를 진행하고, 추출된 Test Case 에 대한 검증을 진행할 예정이다.

이 논문은 2012 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2012-0001845)과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

참고문헌

[1] 정혜정, 정원태, "소프트웨어 신뢰성 테스트를 위한 테스트케이스", 한국멀티미디어학회, Vol.13, No.1, pp.348-351, 2010

[2] Chen Minsong, Qiu Xiaokang, Li Xuandong, "Automatic test case generation for UML activity diagram", Proceedings of the 2006 international workshop on Automation of software test AST, 2006.

[3] 우수정, 김영철, "UML 메카니즘과 원인-결과 다이어그램 기반 테스트케이스 생성을 위한 자동 도구 개발", 한국정보과학회, Vol.39, No.1, pp.205-207, 2012

[4] Gary E. Mogyorodi, "Requirements-Based Testing Cause-Effect Graphing", 2010

[5] 문영준, 박순용, "지능형교통시스템(ITS)의 이해와 동량", 대한전기학회, vol.55, No.11, pp.18-21, 2006

[6] 이영인, "지능형 고시교통관리체계(ATMS)의 시스템 구성 및 제공서비스", 대한전자공학회, Vol.24, No.2, pp.179-187, 1992

[7] 한백전자 기술연구소, "ZigbeX 를 이용한 유비쿼터스 센서 네트워크 시스템", 2008