

ISSN 1598-9798



데이터베이스연구

제28권 제2호 2012년 8월

XML 데이터 저장 기법을 활용한 MICOM 기반의 가전제품 테스트 소프트웨어의 설계 및 구현

Design and Implementation of an Appliance Testing Software based
on MICOM using an XML Data Storage Technique

북경수, 임종태, 김준, 안대영, 유재수

Kyoungsoo Bok, Jongtae Lim, Jun Kim, Daeyoung Ahn, Jaesoo Yoo

데이터베이스 소사이어티
Database Society

사단법인 한국정보과학회

The Korean Institute of Information Scientists and Engineers



XML 데이터 저장 기법을 활용한 MICOM 기반의 가전제품 테스트 소프트웨어의 설계 및 구현

Design and Implementation of an Appliance Testing Software based on MICOM using an XML Data Storage Technique

북경수(kyoungsoo Bok)¹, 임종태(Jongtae Lim)², 김준(Jun Kim)³, 안대영(Daeyoung Ahn)⁴, 유재수(Jaesoo Yoo)⁵

요 약

본 논문에서는 XML 데이터 저장 기법을 활용하여 MICOM 기반의 가전제품에 내장된 임베디드 소프트웨어에 대한 기능 시험을 수행하기 위한 테스트 도구를 제안한다. 제안하는 테스트 도구는 GUI 인터페이스를 통해 테스트케이스 생성 및 실행을 수행할 수 있다. 제안하는 테스트 소프트웨어는 입력값 및 이벤트에 대한 테스트케이스를 모두 생성할 수 있다. 입력 값에 대한 테스트케이스는 가전제품에서 인식하는 입출력 신호를 기반으로 자동화 또는 수작업에 의한 테스트케이스를 생성할 수 있다. 이벤트에 대한 테스트케이스는 이벤트에 대한 생성 시간 및 발생 값을 설정하면서 생성할 수 있다. 테스트 수행 결과는 GUI 인터페이스를 실시간으로 확인 가능하며 검색 기능을 통해 특정 테스트 결과만을 확인할 수 있다.

주제어: XML, MICOM, 임베디드 소프트웨어, 테스트, 테스트케이스 시스템생물학

1 충북대학교 전자정보대학 정보통신공학부, 초빙부교수

2 충북대학교 전자정보대학 정보통신공학부, 박사과정

3 테스트마이다스, 대표이사

4 테스트마이다스, 부사장

5 충북대학교 전자정보대학 정보통신공학부 및 충북BIT연구중심대학육성사업단, 교수, 교신저자

† 이 논문은 2011년도 정부(지식경제부)의 재원으로 연구개발특구지원본부의 지원을 받아 수행된 기초연구사업의 결과임(No. A2011DD005)

+ 논문접수: 2012년 07월 15일, 심사완료: 2012년 08월 19일

Abstract

In this paper, we propose an embedded software testing tool using XML data storage technique to perform functional tests for MICOM based consumer home appliances. The proposed testing tool can generate and execute test cases through GUI interfaces. It can generate both input values and events. The test cases of input values are generated by automatic and manual methods and the test cases of events are generated by setting generation times and values. In the proposed software, the test results can be checked in real-time through GUI interfaces and only the results of a particular test by searches can be seen.

Keywords: XML, MICOM, Embedded Software, Testing, Test Case

1. 서론

항공, 자동차, 철도, 원자력 분야에서 사용되는 임베디드 소프트웨어의 검증 및 신뢰성에 대한 인증이 요구되고 있다. 이에 따라 임베디드 소프트웨어에 대한 테스트를 자동적으로 수행하기 위해 많은 연구들이 진행되고 있다[1, 4]. 일반적으로 임베디드 소프트웨어에 대한 테스트는 테스트케이스 설계 및 생성 모듈과 테스트 수행 모듈을 분리하여 동작된다. 기존에 개발된 대부분의 테스트 도구들은 개발 단계에서 소프트웨어의 자체적인 테스트를 수행하기 위한 제품이다. 실제 소프트웨어는 타켓 환경 즉, 하드웨어에 실제 소프트웨어를 탑재한 환경에서 안정적으로 동작되어야 한다. 그러나 실제 임베디드 시스템은 다양한 주변 기기 및 하드웨어 제약 사항 등으로 인해 개발이 완료된 소프트웨어를 탑재하였을 경우 다양한 문제들이 발생할 수 있다[1, 2, 4]. 따라서 임베디드 소프트웨어 개발이 완료된 후 하드웨어 플랫폼에서 실제 테스트를 수행하기 위한 필요성이 증가되고 있다. 그러나 임베디드 소프트웨어는 임베디드 하드웨어의 고유한 특성으로 인해 소프트웨어와 매우 밀접히 결합되어 있다. 또한, 대부분 이벤트 중심으로 동작하기 때문에 테스트 자동화가 어려운 상황이다[3, 5].

임베디드 시스템의 활용분야가 매우 광범위해지면서 시스템의 품질과 신뢰성 보증을 위한 테스트 과정에 많은 관심이 쏟아지고 있다. 특히, 임베디드 시스템의 블랙박스 테스트 기법에 대한 연구가 활발히 진행되면서 테스트 과정에서 요구되는 모델 작성에 대한 연구들이 진행되고 있다[1]. 일반적인 임베디드 소프트웨어는 모델 기반의 설계 도구의 결과물을 활용하여 수행된다[9, 10]. 모델 기반 설계 도구로 가장 많이 사용되는

UML과 Simulink는 제어 알고리즘과 제어 대상을 함께 모델링하고 시뮬레이션 할 수 있는 환경을 제공한다[6]. REACTIS는 Simulink/Stateflow 모델을 기반으로 테스트케스트를 생성하고 모델 테스트에 따른 입출력 값을 저장해 소스 코드 검증을 위한 테스트슈트(test suite)로 재사용한다. REACTIS는 단위 테스트 및 시스템 테스트가 가능하며 C 코드로부터 test suite 자동 생성 및 실행을 통한 실행 오류 검출한다[7]. R-Bench는 RDL(Requirement Diagram Language) 또는 MATLAB Simulink/Stateflow를 이용한 요구사항 모델링한다. 요구사항을 자동으로 분석하고 테스트 케이스 자동 생성 및 테스트 결과 자동 분석을 수행한다[8].

MICOM 기반의 가전제품에 다양한 기능이 부과됨에 따라 이를 제어하기 위한 제어 소프트웨어 기능도 복잡해지고 있다. 이에 따라 가전제품을 제어하기 위한 소프트웨어에 대한 기능 시험을 효율적이고 저비용을 수행하기 위한 자동화된 테스트 도구가 요구되고 있다. 가전제품을 동작시키기 위한 사용자 입력값과 가전제품 내부에서 발생하는 하드웨어적인 이벤트를 정의하고 이를 기반으로 MICOM 기반의 가전제품에 내장된 소프트웨어의 기능 시험을 수행하기 위한 필요성이 증가되고 있다.

본 논문에서는 MICOM 기반의 가전제품에 내장된 소프트웨어의 기능 시험을 수행하기 위한 테스트 도구를 제안한다. 제안하는 테스트 도구는 사용자 입력값 뿐만 아니라 하드웨어 내부적인 이벤트를 정의하고 이를 기반으로 자동화된 테스트를 수행할 수 있으며 테스트를 위한 모든 정보는 XML 형태로 저장 관리한다. 다수의 테스트케이스를 수행시키고 수행된 결과를 확인할 수 있으며 테스트 수행 과정을 일시정지/재시작, 중

지할 수 있다.

본 논문의 나머지 구성은 다음과 같다. 2장에서는 기존에 제안된 테스트 도구들에 대해 분석하고 3장에서는 제안하는 테스트 소프트웨어에 대해 기술한다. 4장에서는 제안하는 테스트 도구의 개발 내용을 기술한다. 마지막 5장에서는 본 논문의 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

[1]에서는 임베디드 시스템의 블랙박스 테스트를 위해 시스템의 하드웨어 특성을 고려한 테스트 모델을 기반으로 하여 테스트 케이스를 생성하는 기법을 제안하였다. 먼저 시스템에 대한 요구사항을 바탕으로 상세명세서와 다이어그램을 작성하고, 시스템의 기능을 모듈화 할 수 있는 블록을 식별한다. 상태 전이를 기반으로 하여 시스템의 행위를 모델링하여 테스트 모델이 구축되고, 모델로부터 상태 전이 표가 작성된다. 상태 전이 표를 작성하여 구체적인 테스트 케이스를 추출하고 임베디드 시스템의 블랙박스 테스트 수행에 활용한다.

[5]에서는 임베디드 소프트웨어에 내재된 결함 위치와 원인을 추적할 수 있도록 하는 임베디드 소프트웨어 인터페이스 테스트하기 위한 Justitia를 제안하였다. 임베디드 시스템 내의 특정 소프트웨어와 연관된 인터페이스를 테스트하기 위해 다음의 테스트 이슈를 해결하는 인터페이스 정보 추출 기법, 테스트 케이스 선정 기법, 테스트 수행 기법을 제안하였다. 제안하는 테스트 기술은 컴파일러가 생성하는 디버깅 정보와 개발자가 디버깅 목적으로 사용하는 이플래이터를 활용하여 어떻게 인터페이스를 추출할 것인가, 무엇이 인터페이스에서 테스트되어야 하는가, 어떻게 인터페이스에서 결함 판정을 할 것인가

의 핵심 테스트 활동을 해결하였다.

[8]에서는 임베디드 소프트웨어의 개발 및 자동화된 테스트를 위한 통합 솔루션으로 R-Bench를 개발하였다. R-Bench는 요구 사항 작성, 테스트케이스 생성, 테스트케이스를 통한 시스템 테스트, 테스트 결과 분석 및 보고서 작성 등의 기능을 제공한다. R-Bench는 전체적인 테스트 흐름은 REED를 통해 시스템의 요구사항을 정형적인 방법으로 표현하며 TSG를 통해 요구 사항을 입력으로 하여 각종 커버리지를 만족하는 테스트케이스를 자동으로 생성하여 이를 테스트 스크립트라는 형식으로 저장한다. 테스트 스크립트 파일을 입력으로 실제 테스트 대상 시스템을 실행시켜 출력 결과값을 수집할 수 있다.

[11]에서는 임베디드 소프트웨어에 대한 소스 분석하여 테스트케이스를 자동으로 생성하는 CodeScroll를 개발하였다. CodeScroll는 임베디드 소프트웨어를 PC환경에서 시험 가능하도록 소스 변경하고 소스 코드를 분석하여 시험 대상 프로그램의 내부 정보를 추출한다. 분석된 내용을 기반으로 다양한 테스트 데이터 조합하여 임베디드 시스템 환경과 동일한 테스트 케이스를 연속 수행할 수 있다. 또한 장시간 실제 임베디드 시스템이 수행한 경우와 동일한 테스트 환경에서 수행이 가능하다. 자동으로 생성된 테스트케이스를 실행하여 테스트를 수행하고 다양한 에러를 모니터링한다.

UML은 임베디드 시스템의 설계 과정에서 가장 많이 사용되는 시각적인 모델링 언어이다. 최근 UML의 모델링 결과를 활용하여 테스트케이스를 자동 생성하기 위한 연구들이 진행되고 있다. [12]에서는 테스트케이스 모델링의 요구 사항을 만족시키기 위해 클래스 다이어그램(class diagram), 시퀀스 다이어그램(sequence diagram), 활동 다이어그램

(activity diagram)을 확장하고 확장된 UML 다이어그램에 기반하여 임베디드 실시간 소프트웨어에 대한 테스트케이스를 자동 생성하기 위한 기법을 제안하였다. [13]에서는 상태 다이어그램(state diagram)과 활동 다이어그램을 SAD(State-Activity Diagram)이라는 중간 표현 방식을 생성하여 테스트케이스를 생성하는 기법을 제안하였다.

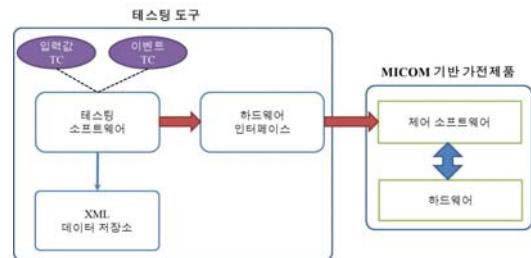
3. MICOM 기반 가전제품 테스트 소프트웨어 설계

3.1 개요

가전제품의 기능이 복잡해지면서 가전제품에 내장된 소프트웨어의 품질을 보장하기 위한 테스트 필요성이 증가되고 있다. 이에 따라 가전제품에 내장된 소프트웨어의 기능 시험을 효율적이고 저비용을 수행하기 위한 자동화된 테스트 도구가 요구된다. 본 논문에서는 XML 기반의 데이터 저장 기법을 활용하여 MICOM 기반의 가전제품에 내장된 소프트웨어에 대한 기능 시험을 수행하기 위한 테스트 도구를 제안한다. 각 모듈들 사이의 연동은 API를 통해 연계할 수 있으며 모든 정보는 XML 저장한다. 사용자에게 의해 수행된 동작은 로그로 저장한다.

그림 1은 제안하는 테스트 소프트웨어의 전체적인 동작 환경을 나타낸 것이다. 가전제품에 대한 기능 시험을 수행하기 위해 입력값 및 이벤트에 대한 테스트케이스를 생성하고 인터페이스 하드웨어를 통해 가전제품에 테스트 데이터를 전송한다. 하드웨어 인터페이스는 가전제품에 인식하는 하드웨어적인 입출력 신호에 따라 가전제품과 통신을 위한 기능을 제공한다. 사용자의 테스트 요청에 따라 하드

웨어 인터페이스는 가전제품에 데이터를 전송하고 처리 결과를 수신하여 테스트 소프트웨어에 전송한다. 이때, 하드웨어 인터페이스는 테스트 소프트웨어에서 전달한 메시지를 가전제품의 제어 소프트웨어에서 인식할 수 있는 제어 신호로 변환하고 제어 소프트웨어로부터 수신한 신호는 소프트웨어가 인식할 수 있는 메시지 규격으로 변환하여 전송한다. 테스트 소프트웨어는 수신한 테스트 결과를 수집하여 테스트 결과를 사용자에게 제공한다.

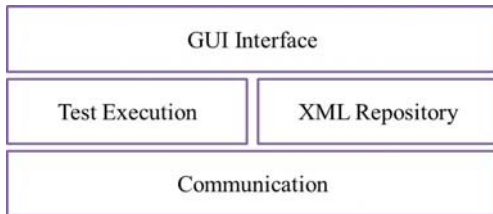


[그림 1] 동작 환경

3.2 시스템 구조

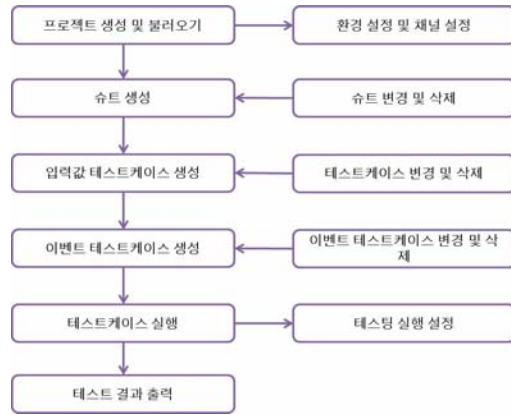
제안하는 테스트 소프트웨어는 그림 2와 같이 4개의 모듈로 구성되어 있다. GUI Interface 모듈은 사용자의 요청에 따라 가전제품에 대한 기능 시험을 수행할 테스트케이스를 생성하고 테스트케이스를 수행시킨다. 수행된 테스트 결과는 실시간적으로 사용자에게 제공되며 완료된 테스트케이스의 결과 목록을 제공한다. Execution 모듈은 GUI를 통해 요청된 테스트케이스를 생성을 수행하고 실제 테스트케이스를 수행하고 제어한다. Execution 모듈은 테스트케이스를 수행시키기 위해 테스트케이스 수행 순서와 시간을 생성하고 스케줄러를 통해 Communication 모듈로 테스트케이스 수행 명령을 전달한다. Communication 모듈은 신호 유형에 따라

테스트케이스 수행 명령을 생성하여 하드웨어로 전달하고 수행된 결과를 수신하여 Repository 저장한다. 또한, 현재 수행되고 있는 테스트케이스에 대한 결과를 GUI 인터페이스를 전송한다. Repository 모듈은 프로젝트 정보, 테스트케이스 정보, 수행 결과 등을 저장하고 읽는 기능을 수행한다. Repository 모듈은 XML 형식으로 모든 정보를 저장하고 XML 문서에서 필요한 정보를 읽는 기능을 제공한다.



[그림 2] 테스트 소프트웨어 구조

제한하는 테스트 소프트웨어를 이용하여 테스트케이스를 생성한 후 실행시키는 수행하는 과정은 다음과 같다. 테스트케이스를 생성하기 위해서는 먼저 프로젝트를 생성한다. 프로젝트를 생성하는 과정에서 하드웨어 정보와 채널을 설정한다. 프로젝트 생성이 완료되면 테스트케이스를 관리하기 위한 슈트를 생성한다. 슈트 생성이 완료되면 슈트 내에 입력값에 대한 테스트케이스를 생성한다. 이벤트에 대한 테스트케이스 생성이 필요할 경우 입력값 테스트케이스 내에 이벤트 테스트케이스를 생성한다. 이벤트 테스트케이스는 필수적인 것은 아니며 필요에 따라 생성한다. 테스트케이스 생성이 완료되면 Setup-Run을 통해 테스트케이스를 실행시키고 실행 결과를 확인한다. 그림 3은 제안하는 테스트 도구의 전체적인 수행 과정을 나타낸 것이다.



[그림 3] 수행 과정

제안하는 테스트 소프트웨어는 입력값에 대한 테스트케이스 InputTC와 이벤트에 대한 테스트케이스 EventTC를 제공한다. 가전제품의 동작은 사용자가 특정 설정값을 선택하여 동작을 수행시킬 수 있으며 가전제품이 동작하는 과정에서 다양한 설정 변환, 새로운 값에 대한 설정 및 추가를 수행할 수 있다. 이와 같이 사용자가 가전제품을 동작시키기 위해 초기에 설정하는 값에 대한 테스트케이스를 InputTC라 하며 가전제품이 동작하는 과정에서 새로운 값의 설정 및 변환에 대한 테스트케이스를 EventTC라고 한다. 즉, EventTC는 가전제품의 동작 중 사용자 또는 하드웨어에서 발생하는 다양한 이벤트들에 대한 테스트케이스를 의미한다.

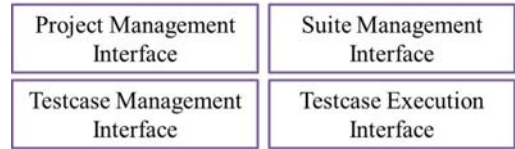
3.3 세부 모듈

(1) GUI Interface 모듈

GUI Interface는 사용자의 요청에 따라 프로젝트 생성부터 테스트케이스를 생성 및 실행을 위한 기능을 제공한다. 사용자의 요청에 따라 필요한 정보를 저장, 실행, 출력하는 기능을 수행하며 수

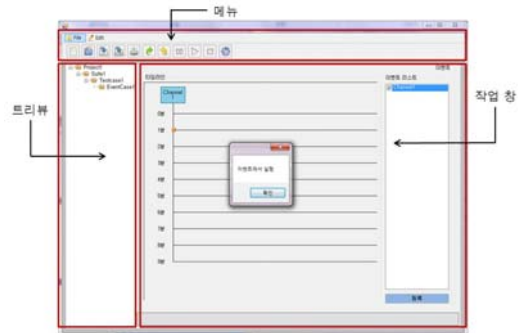
행 결과는 XML Repository에 저장된다. GUI Interface에서는 기능 시험을 필요한 모든 정보는 XML Repository에 저장하고 프로젝트 불러오기를 수행할 경우 GUI 화면 구성을 위한 인스턴스(instance)를 생성하고 출력한다. GUI Interface에서는 프로젝트 정보에 대한 생성, 변경, 저장, 스위트 정보에 대한 생성, 추가, 변경, 삭제, 테스트케이스에 대한 생성, 추가, 변경, 삭제, 테스트케이스에 대한 수행 및 결과 출력 기능을 제공한다.

그림 4는 GUI Interface에 대한 세부 모듈은 나타낸 것이다. Project Management Interface는 사용자의 요청에 따라 프로젝트를 생성하고 기능 시험을 수행할 가전제품에 대한 환경 설정 및 입출력 신호를 설정한다. 입출력 신호는 가전제품에서 인식할 수 있는 하드웨어적인 신호를 사용자 관점에서 인식할 수 있도록 저장하는 기능을 제공하며 이를 채널이라고 한다. 일반적으로 가전제품에 대한 기능 시험을 위해 다수의 테스트케이스를 생성하고 실행한다. 다수의 테스트케이스를 존재할 때 관련된 테스트케이스를 그룹화하여 관리하기 위해 스위트(suite)를 생성한다. Suite Management Interface는 스위트를 관리하는 기능을 수행한다. 또한, 스위트 내에 생성된 테스트케이스를 생성하고 관리하는 기능을 제공한다. Testcase Management Interface는 스위트 내에 테스트케이스를 생성하고 관리하는 기능을 제공한다. 제안하는 테스트 도구에서는 InputTC와 EventTC를 생성할 수 있으며 생성된 테스트케이스는 변경, 삭제가 가능하다. Testcase Execution Interface는 가전제품에 대한 기능 시험을 수행하기 위해 테스트케이스를 실행시켜 실행 결과를 확인하는 기능을 제공한다. 또한, 수행 중인 테스트 과정을 일시정지/재시작, 중지할 수 있는 기능을 제공한다.



[그림 4] GUI Interface 세부 모듈

제안하는 테스트 도구의 GUI 인터페이스는 그림 5와 같이 세 부분으로 구성되어 있다. 상단에 메뉴는 테스트 도구를 사용하기 위한 메뉴 목록과 실행 아이콘으로 구성되어 있다. 왼쪽에 트리뷰는 프로젝트 내에 생성된 스위트, 테스트케이스의 목록을 계층적으로 관리하여 출력한다. 화면의 오른쪽에 존재하는 작업 창은 프로젝트 관리, 스위트 관리, 테스트케이스 관리 및 실행을 실제로 수행하고 그 결과를 출력한다.



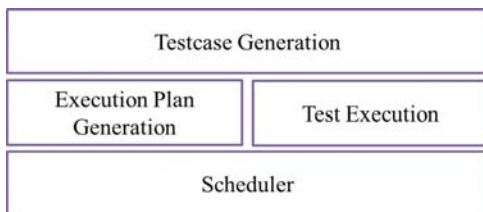
[그림 5] GUI Interface 화면

(2) Test Execution 모듈

Test Execution은 GUI 요청에 따라 실제 테스트케이스를 생성하고 실행하는 기능을 수행한다. Test Execution은 테스트케이스 실행 요청에 따라 테스트를 수행할 순서와 시간을 설정하고 스케줄러를 실행시켜 Communication 모듈을 데이터를 전송한다. Test Execution에서는 입력값에 대한 자동화된 테스트케이스 생성, 테스트 수행 요청에

따른 스케줄러 생성, 테스트 수행 과정을 제어한다. 테스트 수행 과정에 대한 제어는 스케줄러를 일시중지/재시작, 중지하는 역할을 수행한다.

그림 6은 Test Execution의 세부 모듈을 나타낸 것이다. Testcase Generation은 테스트케이스를 생성하는 기능을 수행한다. Testcase Generation에서는 InputTC와 EventTC를 생성할 수 있다. InputTC는 MT와 AT 두가지 유형의 테스트케이스 생성 유형을 제공한다. AT는 가전제품을 동작시키기 위해 사용자가 입력 가능한 유형을 조합하여 자동적으로 테스트케이스를 생성하며 MT는 사용자가 선택한 특정 입력값에 대한 테스트케이스만을 생성한다. Test Execution은 실제 테스트를 동작시키고 제어하는 기능을 수행한다. Test Execution은 테스트 수행 요청에 따라 스케줄러를 구동시키고 필요에 따라 테스트 동작 과정에 대해 일시정지/재시작, 중지하는 기능을 제공한다. Execution Plan Generation은 테스트케이스를 실행시키기 위한 순서를 설정하는 것으로 테스트 수행 요청이 전달되면 InputTC와 EventTC에 대한 수행 시간을 생성하여 스케줄러에 전달한다. 스케줄러는 Execution Plan Generation에서 생성된 테스트 수행 순서에 따라 테스트를 수행할 메시지를 Communication으로 전달한다. 스케줄러는 테스트에 대한 일시정지/재시작, 중지 요청에 따라 테스트 과정을 제어하는 역할을 수행한다.



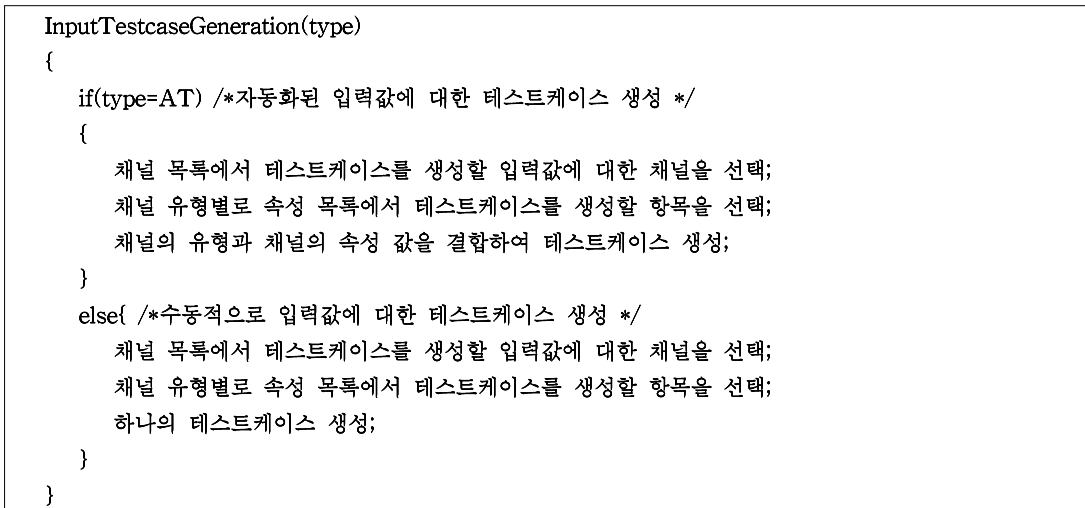
[그림 6] Test Execution 세부 모듈

그림 7은 InputTC를 생성하는 InputTestcaseGeneration() 알고리즘을 나타낸다. InputTC는 생성하는 방법은 AT와 MT 두 가지 방법이 존재한다. AT는 입력값 유형을 결합하여 입력값에 대한 테스트케이스를 생성하는 방법으로 채널 목록에서 테스트케이스를 생성할 입력값과 유형에 대한 항목을 조합하여 테스트케이스를 생성한다. MT는 채널 목록에서 선택한 특정 항목에 대한 테스트케이스만 수작업으로 생성하는 방법이다. 그림 8은 EventTC를 생성하는 EventTestcaseGeneration() 알고리즘을 나타낸다. 이벤트에 대한 테스트케이스를 생성하기 위해서는 채널 목록에서 이벤트에 대한 채널을 선택하고 이벤트에 대한 발생 시간 및 발생 값을 설정한다.

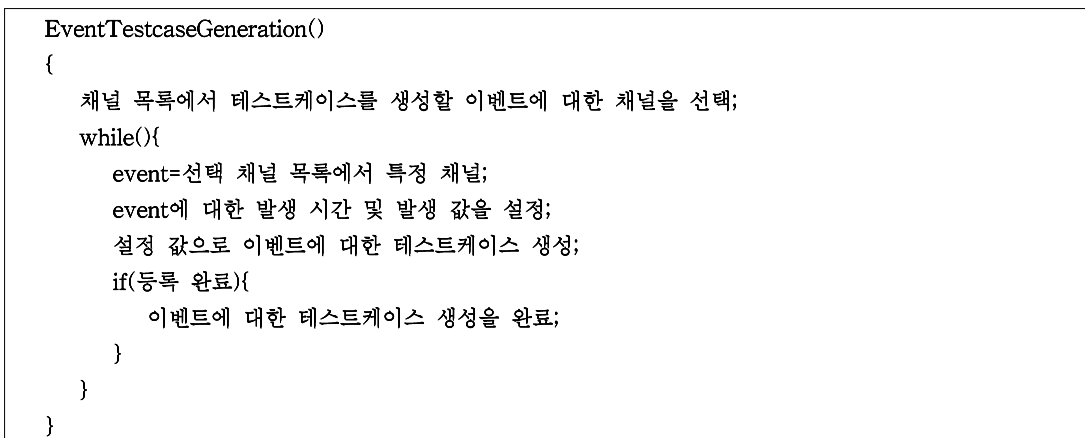
(3) Communication 모듈

Communication은 하드웨어 인터페이스를 통해 가전제품에 테스트 메시지를 전송하고 수신하는 기능을 수행한다. 테스트 실행에 따라 스케줄러는 Communication에 테스트 수행 메시지를 전달하고 Communication은 하드웨어 인터페이스를 통해 테스트 메시지를 가전제품으로 전달한다. 가전제품에 테스트를 수행할 결과는 하드웨어 인터페이스를 통해 Communication 모듈로 전달된다.

그림 9는 Communication의 세부 모듈을 나타낸 것이다. Send Message는 신호 유형에 따라 전달할 메시지를 규격을 생성하고 하드웨어 인터페이스를 통해 가전제품으로 테스트 메시지를 전달한다. Receive Message는 하드웨어 인터페이스를 통해 가전제품의 테스트 수행 결과를 수신하고 수신된 내용을 XML Repository에 저장한다. Message Daemon은 하드웨어 인터페이스를 메시지를 송수신하기 위한 메몬 프로세스로 Send



[그림 7] 입력값에 대한 테스트케이스 생성



[그림 8] 이벤트에 대한 테스트케이스 생성

Message와 Receive Message 요청에 따라 데이터를 송수신한다.



[그림 9] Communication 세부 모듈

(4) XML Repository 모듈

제안하는 테스트 도구에는 가전제품에 대한 기능 시험을 위한 모든 정보는 XML 형식으로 저장한다. XML Repository는 프로젝트, 슈트, 테스트케이스, 테스트 수행 결과를 XML 형식으로 저장하고 읽는 기능을 제공한다. 프로젝트가 생성된 XML Repository는 프로젝트 이름으로 서브디렉

램의 대부분의 작업이 이루어집니다. 사용자는 메뉴나 단축 아이콘들을 이용하여 프로그램의 기능을 수행할 수 있다.

표 1 시스템 사양

항목	최소 사양
CPU	펜티엄 5 이상
메모리	2GB 이상
OS	MS Windows XP 이상
필수 소프트웨어	MS .NET Framework 4 이상

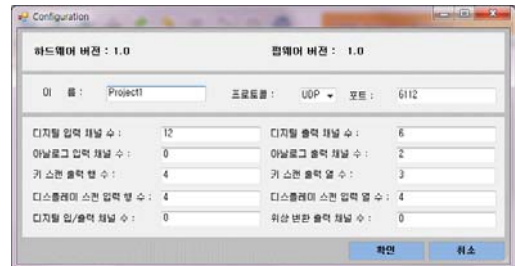
본 논문에서 제안하는 테스트 도구는 프로젝트를 생성하고 프로젝트 단위로 테스트케이스 관리 및 수행이 가능하다. 제안하는 테스트 도구는 프로젝트 관리, 슈트 관리, 테스트케이스 생성 및 실행 기능을 제공한다. 프로젝트 관리 기능에서는 프로젝트 생성 및 채널 관리 기능을 제공한다. 슈트 관리 기능에서는 슈트 생성, 변경, 삭제 기능을 제공하며 슈트 내에 다수의 테스트케이스를 생성하고 수행시킬 수 있다. 테스트케이스 생성 및 실행에서는 입력값에 대한 테스트케이스를 생성, 변경, 삭제하는 기능과 생성된 테스트케이스를 실행시키고 실행 결과를 확인할 수 있는 기능을 제공한다.

4.2 구현 결과

(1) 프로젝트 생성

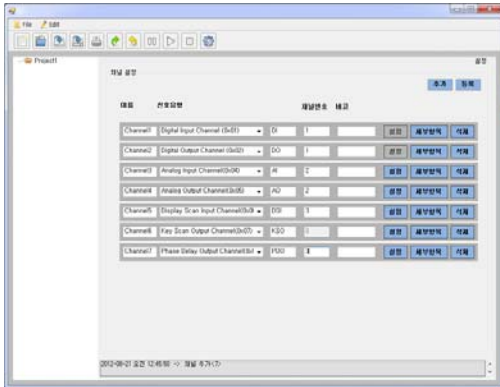
테스트케이스를 생성하고 실행시키기 위해서는 프로젝트를 생성한다. 즉, 프로젝트는 테스트케이스를 실행하기 위해 최상위에 생성하는 루트를 의미하며 테스트케이스를 실행하기 위해서 가장 처

음으로 생성한다. 프로젝트 하위에 슈트를 생성하여 테스트케이스를 관리할 수 있다. 메뉴 또는 단축 아이콘을 이용하여 프로젝트를 생성할 때 사용자는 프로젝트 이름과 프로젝트에 필요한 하드웨어 정보를 설정합니다. 그림 12는 프로젝트 생성 화면을 나타낸 것이다.



[그림 12] 프로젝트 생성

가전제품은 다양한 유형의 입출력 신호들이 존재한다. 테스트케이스를 생성하여 실행시키기 위해서는 실제 가전제품에서 인식할 수 있는 신호로 데이터를 전달해야 합니다. 채널 등록은 각각의 입력케이스나 이벤트가 동작하는 신호유형과 여러 가지 항목들을 등록한 것을 의미한다. 채널은 실제적인 하드웨어에 대한 정보도 포함하고 있기 때문에 실제 하드웨어와 다른 신호유형이나 항목으로 채널을 설정할 경우에 테스트케이스가 정확하게 동작하지 않을 수 있습니다. 따라서 채널을 등록할 때 하드웨어의 신호유형과 항목들을 일치하게 입력한다. 프로젝트가 생성하면 사용자는 채널 정보들을 설정한다. 그림 13은 프로젝트 생성 과정에서 채널 정보를 등록하는 화면을 나타낸 것이다. 채널 등록은 추가 버튼을 클릭하여 새로운 채널을 추가할 수 있다.



[그림 13] 채널 등록

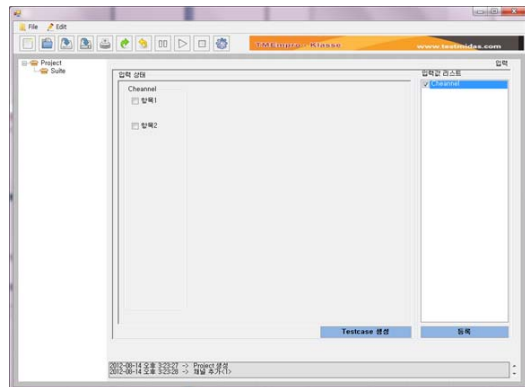
추가 버튼을 클릭하여 채널을 추가하면 채널에 대한 세부 내용을 설정한다. 사용자는 채널이름, 채널유형, 채널번호, 기타정보 및 세부항목을 설정한다. 채널 신호 유형은 9가지가 존재하며 채널을 선택한 후 설정 버튼을 클릭하여 각각의 채널 유형에 해당하는 기타정보를 설정한다. 세부 항목 버튼을 클릭하여 채널에서 선택할 수 있는 세부 정보를 설정한다. 그림 14는 채널의 세부 항목을 설정하는 화면이다.



[그림 14] 세부 항목 설정

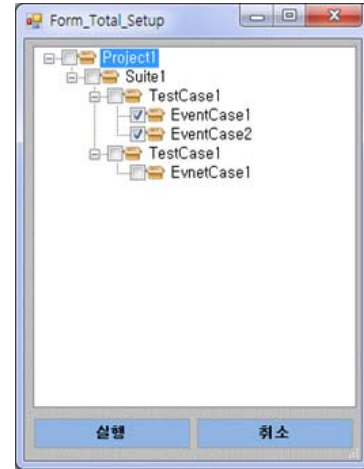
(2) 테스트케이스 생성 및 관리

슈트는 프로젝트 내에 존재하는 테스트케이스를 관리하기 위한 단위로 하나의 슈트 내에는 다수의 테스트케이스를 생성하고 관리할 수 있습니다. 테스트케이스를 생성하기 전에 슈트를 먼저 생성한다. 가전제품은 일반 사용자가 설정 가능한 다양한 값들 존재한다. Create-TC는 사용자 입력값에 대한 테스트케이스를 직접 생성한다. Create-TC를 수행하기 위해서는 생성한 테스트케이스에 대한 이름을 설정하고 생성할 입력값을 설정한다. Autogen-TC는 사용자 입력값에 대해 조합 가능한 모든 경우 수를 생성하여 테스트케이스를 생성한다. 그림 15는 Autogen-TC를 수행하는 화면을 나타낸다. 사용자는 프로젝트 생성 과정에서 설정한 채널 목록에서 입력값으로 사용할 채널을 선택한다. 목록에서 입력값으로 사용할 채널을 선택하면 각 채널에 대해 테스트케이스로 조합할 채널의 세부 항목을 선택한다. 'Testcase 생성' 버튼을 클릭하면 각 채널의 세부 항목들을 모두 조합하여 InputTC를 자동으로 생성한다.

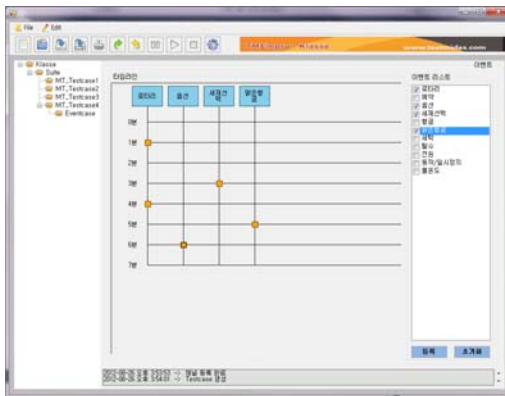


[그림 15] 자동화된 입력값에 대한 테스트케이스 생성

가전제품은 특정 입력값에 대해 동작하는 과정에서 다양한 이벤트들이 발생할 수 있다. Create-EC는 특정 입력값에 대한 테스트케이스에서 EventTC를 생성한다. 그림 16은 EventTC를 생성하는 화면이다. EventTC를 생성하기 위해서는 채널 목록에서 EventTC로 생성할 채널을 선택한 후 실제 이벤트로 발생시킬 값과 시간을 설정한다. 이벤트로 발생시킬 수 있는 값은 사용자가 입력값 또는 하드웨어에서 발생하는 이벤트를 의미한다. 이벤트의 세부 정보는 타임라인이나 버튼을 클릭하여 입력한다.



[그림 17] 테스트케이스 실행



[그림 16] 이벤트에 대한 테스트케이스 생성

(3) 테스트 수행 및 결과

프로젝트 내에서 생성된 다수의 테스트케이스가 생성되어 있다. 프로젝트 내에 생성된 다수의 테스트케이스를 수행시키기 위해 실행시킬 테스트케이스를 선택한다. Setup-Run은 프로젝트 내에 생성된 테스트케이스를 선택하여 테스트를 실행시킬 수 있다. 그림 17은 테스트케이스를 실행시키기 위해 작성된 테스트케이스를 선택하는 화면이다.

현재 수행 중인 테스트에 대한 시험 결과를 실시간으로 확인할 수 있도록 테스트 수행 결과 출력 기능을 제공한다. 그림 18은 결과 출력을 위한 화면을 나타낸다. Setup-Run을 수행하여 테스트케이스를 수행하면 테스트 수행 결과 출력 화면에 출력한다. 검색 버튼을 클릭하여 검색 조건을 입력할 경우 특정 테스트케이스에 대한 수행 결과만 확인할 수 있다.

No.	Date	Time	Testcase	Trace	Channelname	Channelnumber	Message
1	2011년 08월 27일	오후 3:03:16	testcase_0	out	통신	0	?
2	2011년 08월 27일	오후 3:04:20	testcase_0	out	통신	0	Ack with no error
3	2011년 08월 27일	오후 3:04:20	testcase_0	out	세탁	0	분쇄작
4	2011년 08월 27일	오후 3:04:42	testcase_0	out	세탁	0	Ack with no error
5	2011년 08월 27일	오후 3:04:50	testcase_0	out	탈수	0	옴
6	2011년 08월 27일	오후 3:04:50	testcase_0	out	탈수	0	Ack with no error
7	2011년 08월 27일	오후 3:05:09	testcase_0	out	물중도	0	물중도
8	2011년 08월 27일	오후 3:05:40	testcase_0	out	물중도	0	Ack with no error
9	2011년 08월 27일	오후 3:06:40	testcase_0	out	물차/물시감지	0	물차/물시감지
10	2011년 08월 27일	오후 3:06:50	testcase_0	out	물차/물시감지	0	Ack with no error

[그림 18] 테스트 수행 결과

5. 결론 및 향후 연구

본 논문에서는 MICOM 기반의 가전제품에 내장된 임베디드 소프트웨어에 대한 테스트하기 위한 테스트 도구들을 제안하였다. 제안하는 테스트 도구는 가전제품에 대한 사용자 입력값과 이벤트에 대한 테스트케이스를 생성하고 다수의 테스트 케이스를 수행시킬 수 있다. 제안하는 테스트 도구에서는 테스트케이스 생성 및 실행에 필요한 모든 정보를 XML 형식으로 저장한다. 제안하는 테스트 도구는 사용자가 테스트케이스 생성 및 실행으로 보다 효과적으로 수행할 수 있는 GUI 인터페이스를 제공한다. 향후 연구로 이벤트에 대한 자동화된 테스트케이스 생성 방법에 대한 연구를 진행할 예정이며 테스트 수행 결과에 보고서 작성 기능을 연구할 예정이다.

6. 참고문헌

- [1] 정소영, 장영원, 유철중, “임베디드 시스템을 위한 상태 전이 모델 기반 테스트 케이스 생성 기법”, 한국정보기술학회논문지, 제9권, 제4호, pp.11-21, 2011
- [2] 최유나, 서주영, 최병주, “임베디드 소프트웨어를 위한 테스트와 디버깅 연계 자동화 방안”, 정보과학회논문지 : 컴퓨팅의 실제 및 레터, 제16권 제5호, pp.576-580, 2010
- [3] 최병주, 서주영, “임베디드 소프트웨어 테스트 전략 및 실제”, 전자공학회지, 제39권, 제1호, pp.17-23, 2012
- [4] 배현섭·윤광식·오승욱, “임베디드 소프트웨어 테스트 이슈 및 현황”, 정보과학회지, 제24권, 제8호, pp.40-45, 2006
- [5] 서주영, 최병주, “임베디드 소프트웨어 테스트를 개선하기 위한 에뮬레이터 기반 인터페이스 테스트 도구”, 정보과학회논문지 : 컴퓨팅의 실제 및 레터, 제14, 제6권, pp.547-558, 2008
- [6] Y. Zhan, J. A. Clark, “The State Problem for Test Generation in Simulink”, Proc. Genetic and Evolutionary Computation Conference, pp.1941-1948, 2006
- [7] S. Sims, D. C. DuVarney, “Experience Report: The Reactis Validation Tool”, Proc. ACM SIGPLAN International Conference on Functional Programming, pp.137-140, 2007
- [8] R-Bench, <http://www.btstech.co.kr>
- [9] P. Sampath, A. C. Rajeev, S. Ramesh, K. C. Shashidhar, “Testing Model-Processing Tools for Embedded Systems”, Proc. IEEE Real-Time and Embedded Technology and Applications Symposium, pp.203-214, 2007
- [10] P. Iyengar, “Test Framework Generation for Model-Based Testing in Embedded Systems”, EUROMICRO Conference on Software Engineering and Advanced Applications, pp.267-274, 2011
- [11] CodeScroll, <http://suresofttech.com/>
- [12] Yin Yongfeng, Liu Bin, Lu Minyan, Li Zhen, “Test Cases Generation for Embedded Real-time Software Based on Extended UML”, Proc. International Conference on Information Technology and Computer Science, pp.69-74, 2009
- [13] S. K. Swain, D. P. Mohapatra, and R.

Mall, "Test Case Generation Based on State and Activity Models", Journal of Object Technology, Vol.9, No.5, pp.1-27, 2010



복 경 수

1998년 충북대학교 수학과 (이학사)

2000년 충북대학교 정보통신 공학과 (공학석사)

2005년 충북대학교 정보통신

공학과 (공학박사)

2005년 - 2008년 한국과학기술원 정보전자연구소

2008년 - 2011년 가인정보기술 연구소 연구원.

2011년 - 현재 충북대학교 전자정보대학 정보통신공학부 초빙교수.

관심분야 : 데이터베이스 시스템, 자료저장시스템, 이동 객체 데이터베이스, RFID 및 센서네트워크, 모바일 P2P 네트워크



임 중 태

2009년 충북대학교 정보통신공학과 (공학사)

2011년 충북대학교 정보통신공학과 (공학석사)

2011년 - 현재 충북대학교 정

보통신공학부 박사과정

관심분야 : 정보검색, 위키기반서비스, 분산 데이터베이스



김 준

1983년 부산대학교 계산통계학과(공학사)

1986년 KAIST 전산학과(공학석사)

2011년 충북대학교 멀티미디어

어공학과(공학박사)

1986년 - 2011년: ETRI 저장시스템연구팀

2008년 - 현재: (주)테스트마이다스 대표이사

관심분야 : SW 테스트, 파일 시스템, DBMS



안 대 영

1984년 한양대학교 전자공학과 (공학사)

1986년 KAIST 전기 및 전자공학과 (공학석사)

1999년 KAIST 전기 및 전자공

학과 (공학박사)

1986년 - 2002년 ETRI 팀장

2002년 - 2004년 서원대학교 컴퓨터교육과 교수

2005년 - 2007년 육성전자 연구소 그룹장

2008년 - 현재 : (주)테스트마이다스 부사장

관심 분야 : 시스템소프트웨어, 실시간 운영체제, 소프트웨어 테스트



유 재 수(교신저자)

1989년 전북대학교 컴퓨터공
학과 (공학사)

2007년 한국과학기술원 전산
학과 (공학석사)

2011년 한국과학기술원 전산

학과 (공학박사)

1995년 - 1996년 목포대학교 전산통계학과 (전임강사)

1996년 - 현재 충북대학교 전자정보대학 정보통신공학
부 및 충북BIT연구중심대학육성사업단 교수.

관심분야: 데이터베이스 시스템, 정보검색, 센서 네트워
크 및 RFID, 멀티미디어 데이터베이스, 분산 객체 컴퓨
팅, 바이오인포메틱스 등

E-mail : yjs@chungbuk.ac.kr