

컴포넌트 기반 로봇 소프트웨어의 지속적인 통합 및 테스트를 위한 프레임워크

최형섭, 강정석, 맹상우, 박홍성
강원대학교

An Architecture of Testing Automation Framework for Component-based Robot Software

Hyeong-Seob Choi, Jeong-Seok Kang, Sang-Woo Maeng, Hong-Seong Park
Kangwon National University

Abstract - 본 논문에서는 로봇 소프트웨어 컴포넌트 테스트 자동화에 적합한 테스트 자동화 프레임워크 및 계층적 로봇 소프트웨어 테스트 모델을 제안한다. 테스트 자동화 프레임워크는 로봇 소프트웨어 컴포넌트의 지속적인 통합, 다양한 로봇 환경 및 분산 컴포넌트 환경을 지원한다. 테스트 자동화 프레임워크를 실제 구현하여 테스트 베드에 설치 및 평가를 통하여, 본 논문에서 제안하는 프레임워크가 효율적임을 보인다.

1. 서 론

현재 일반적인 소프트웨어 테스트 자동화 도구에 대한 연구는 90년대 이후로 활발히 진행되고 있다. 소프트웨어 자동 테스트 도구의 종류는 테스트 프로세스의 지원 범위에 따라서 테스트 수행의 자동화 도구, 테스트 케이스의 자동생성 도구, 테스트 관리의 자동화 도구로 분류된다 [1]. 그러나 기존 연구들[3-5]은 로봇 소프트웨어 컴포넌트를 테스트 하는데 다음과 같은 고려사항이 부족하다.

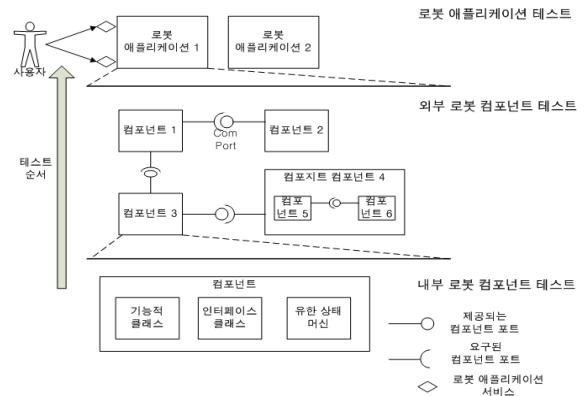
- 첫째, 지속적인 통합 - 로봇 시스템은 위치 인식, 사용자 인지, 주행 기술 등 여러 기술을 필요로 하므로 다양한 로봇 개발자들에게 의해 개발된다. 개발된 로봇 소프트웨어 컴포넌트는 지속적으로 통합 및 빌드됨으로써 로봇 소프트웨어 컴포넌트통합 시 발생할 수 있는 결함을 조기에 발견 할 수 있어야 한다[2].
- 둘째, 다양한 로봇 환경 - 로봇 소프트웨어 컴포넌트는 로봇의 요구 사항에 따라 다양한 하드웨어 플랫폼 및 소프트웨어 플랫폼 상에서 동작한다. 소프트웨어 자동 테스트 도구는 이기종의 플랫폼 상에서 동작하는 로봇 소프트웨어 컴포넌트에 대해 이식성(portability), 신뢰성(reliability) 등을 테스트 할 수 있어야 한다.
- 셋째, 분산 컴포넌트 환경 - 로봇 소프트웨어 컴포넌트는 기능에 따라 여러 환경에 분산/배치 될 수 있으며 미들웨어 기술을 통해서도 통신을 한다. 소프트웨어 자동 테스트 도구는 다양한 환경에 분산된 컴포넌트들간의 상호운용성 테스트를 할 수 있어야 한다.
- 넷째, 다양한 개발 언어 - 로봇 소프트웨어 컴포넌트는 요구사항에 따라 다양한 언어(C, C++, Java 등)로 개발 될 수 있다.

본 논문에서는 위와 같은 고려사항을 만족하는 컴포넌트 기반 로봇 소프트웨어의 지속적인 통합 및 계층적 테스트 자동화를 위한 분산 프레임워크 구조를 제안한다. 제안한 테스트 자동화 분산 프레임워크는 개발자 또는 테스터가 쉽게 로봇 소프트웨어 컴포넌트를 테스트 할 수 있도록 하기 위한 웹 인터페이스, 개발된 컴포넌트를 지속적으로 통합하고 각 컴포넌트들에 대한 테스트 슈트를 자동 생성 및 관리하는 테스트 엔진 서버, 지속적으로 로봇 컴포넌트의 소스를 통합하기 위한 버전 관리 서버, 그리고 다양한 로봇 환경에서 컴포넌트를 빌드/실행 하고 테스트 결과를 테스트 엔진 서버로 전달하는 경량 빌드/실행 에이전트로 구성된다.

본 논문은 다음과 같이 구성된다. 제 2장에서는 로봇 소프트웨어의 계층적 테스트 모델에 대하여 기술하며, 제 3장에서는 본 논문에서 제안하는 분산 테스트 자동화 프레임워크에 대하여 기술한다. 제 4장에서는 제안한 테스트 자동화 프레임워크의 동작성을 검증하기 위한 실험과 정성적 평가에 대해 설명하고 마지막으로 제 5장에서 결론을 맺도록 한다.

2. 로봇 소프트웨어의 계층적 테스트 모델

컴포넌트 기반 로봇 소프트웨어는 단일 컴포넌트 또는 여러 컴포넌트로 구성되어 위치 인식, 사용자 인지 서비스 등을 제공하며 각 소프트웨어 컴포넌트는 내부에 기능적 클래스, 하위 컴포넌트 또는 통신 미들웨어와 통신을 위한 인터페이스, 그리고 유한 상태 머신을 가지고 있다. 로봇 소프트웨어의 신뢰성을 보장하기 위해서는 우선 로봇 소프트웨어를 구성하는 각 컴포넌트들의 신뢰성을 보장하여야 한다.



〈그림 1〉 로봇 소프트웨어의 계층적 테스트 모델

본 장에서는 로봇 소프트웨어의 계층적 테스트 모델에 대해 기술한다. 제안된 계층적 테스트 모델은 그림 1에서 보는바와 같이 내부 로봇 컴포넌트 테스트, 외부 로봇 컴포넌트 테스트, 그리고 로봇 애플리케이션 테스트로 이루어져 있다.

테스트 자동화 프레임워크는 이 모델을 이용하여 로봇 소프트웨어 컴포넌트에 대한 테스트를 수행한다.

3. 테스트 자동화 프레임워크 구조

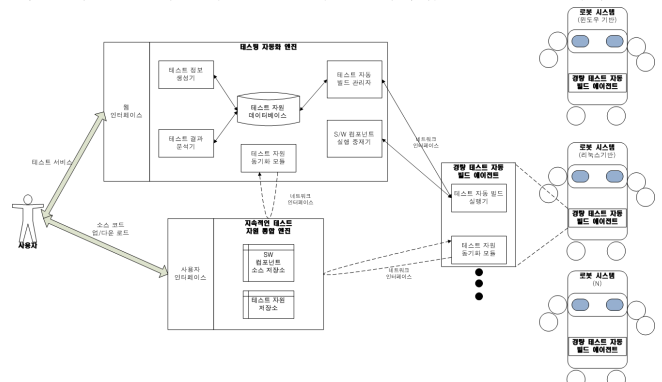
3.1 테스트 자동화 프레임워크 개요

본 장에서는 제 2장에서 정의한 계층적 테스트 모델을 기반으로 로봇 소프트웨어 컴포넌트에 대해 테스트를 자동으로 수행하는 프레임워크의 구조 및 핵심 모듈의 동작에 대해 기술한다.

테스트 자동화 프레임워크의 구조는 그림 2에서 보는 바와 같이 웹 사용자 인터페이스, 테스트 엔진 서버, 로봇 컴포넌트 버전 관리 서버, 그리고 경량 빌드/실행 에이전트로 구성된다.

웹 사용자 인터페이스는 테스트 슈트 관리, 테스트 결과 분석, 로봇 컴포넌트 코드 분석, 테스트 드라이버 또는 로봇 컴포넌트 자동 빌드 및 실행 인터페이스 등이 있다. 사용자(개발자 혹은 테스터)는 웹 브라우저를 통해 웹 사용자 인터페이스에 접속할 수 있다.

테스트 엔진 서버는 테스트 자동화 프레임워크의 핵심 요소로서 로봇 소프트웨어 컴포넌트에 대해 테스트 슈트(테스트 케이스, 테스트 드라이버)를 자동 생성하고 다양한 환경에 분산되어 있는 빌드/실행 에이전트



〈그림 2〉 로봇 소프트웨어 테스트 자동화 프레임워크 구조

를 이용하여 테스트 슈트를 실행하고, 실행 결과를 모니터링 하는 역할을 담당한다.

로봇 컴포넌트 버전 관리 서버는 컴포넌트 소스의 지속적인 통합을 위한 핵심 요소로서 사용자는 버전 관리 클라이언트를 통해 개발된 컴포넌트의 소스를 업/다운로드 할 수 있어 다른 로봇 소프트웨어 개발자의 컴포넌트 소스와 지속적으로 통합된다. 로봇 컴포넌트 버전관리 서버 내에 존재하는 로봇 컴포넌트 소스 저장소는 테스트 엔진 서버와 서로 공유되어 테스트 슈트 생성 및 실행 시 이용된다.

마지막으로 경량 빌드/실행 에이전트는 실제 로봇 컴포넌트가 실행되는 다양한 로봇 환경에 내장되어 테스트 엔진 서버로부터 테스트 슈트를 받아 빌드/실행하며 수행 결과를 업로드하는 역할을 담당한다.



<그림 3> 테스트 시스템 구성도

3.2 테스트 자동화 프레임워크 동작 흐름

테스팅 자동화 프레임워크의 원활한 동작에 있어 각 각의 내부 모듈들은 직접적으로 데이터를 주고 받지 않고 서버 저장소들(테스트 자원 저장소, 서버 데이터베이스, 로봇 컴포넌트 소스 저장소)을 통해 데이터를 주고받으므로 테스트 엔진 서버의 내부 모듈들과 서버 저장소들, 빌드/실행 에이전트간의 상호작용은 중요하다.

3.2.1 로봇 컴포넌트 테스트 케이스 자동 생성

로봇 컴포넌트 테스트 케이스 생성기는 앞서 언급한 바와 같이 텍스트 형식의 테스트 케이스 파일 및 XML 형식의 환경 설정 파일을 생성한다. 사용자에게 의해 테스트할 로봇 소프트웨어 컴포넌트와 테스트 기법(경계값 분석, 동등 분할, 조합 등)이 정해지면 로봇 컴포넌트 저장소로부터 컴포넌트의 소스 정보를 읽어 선택된 테스트 기법에 맞게 테스트 케이스를 생성한 후 테스트 드라이버 코드 생성을 위해 테스트 환경 설정 파일을 생성한다. 생성된 테스트 케이스 파일과 환경 설정 파일은 테스트 자원 저장소에 저장되고 각 파일들의 정보는 서버 데이터베이스에 저장한다.

3.2.2 테스트 드라이버 코드 자동 생성

테스트 드라이버 코드 생성기는 서버 데이터베이스에서 테스트 케이스 환경 파일 정보를 읽어 테스트 자원 저장소로부터 테스트 케이스 환경 파일(클래스명, 메소드 명, 메소드 파라미터 정보 등)을 읽어 XML 파서를 통해 분석한다. 분석한 정보를 통해 테스트 커버리지 계산이 필요할 경우 테스트 커버리지 코드 삽입기를 이용하여 로봇 컴포넌트 소스 코드에 테스트 커버리지 계산을 위한 코드를 삽입한 후 테스트 자원 저장소에 테스트 드라이버 코드와 함께 저장한다. 생성된 테스트 드라이버의 정보는 서버 데이터베이스에 저장한다.

3.2.3 테스트 드라이버 코드 자동 빌드

테스트 엔진 서버의 자동 빌드 관리는 사용자가 빌드 시간 및 빌드 환경 파일을 저장하면 내부 스케줄러를 이용하여 빌드 시간을 예약하고 빌드 환경 정보를 서버 데이터베이스에 저장한다. 스케줄러에 의해 빌드 시간이 되었다는 이벤트를 수신하면 TCP/IP를 통해 로봇 시스템상에 존재하는 빌드 에이전트에 빌드 환경파일 명과 함께 빌드를 수행하라는 명령을 전송한다. 빌드 에이전트는 빌드 명령을 수신하면 테스트 자원 저장소로부터 해당 빌드 환경 파일을 다운로드 한 후 XML 파서를 통해 분석한다. 빌드에 필요한 테스트 드라이버 코드 및로봇 컴포넌트 소스만을 다운로드 하여 빌드를 수행한 후 빌드 로그 파일과 빌드가 성공하였다면 테스트 드라이버의 실행파일을 테스트 자원 저장소에 업로드한다. 그런 다음 빌드의 결과를 테스트 엔진 서버의 빌드 관리기에 전송한다.

3.2.4 테스트 드라이버 코드 자동 실행

테스트 엔진 서버의 자동 실행 관리는 자동 빌드 관리기와 마찬가지로 내부 스케줄러를 이용하여 실행 시간을 예약하고 환경 정보를 서버 데이터베이스에 저장한다. 이 과정에서 중요한 동작은 테스트 드라이버 코드를 실행 시 필요한 로봇 컴포넌트의 DLL 파일이 존재 할 시 로봇 컴포넌트 소스 저장소에서 다운로드 한다는 것이다. 또한 테스트 드라이버를 실행 한 후 다양한 로그 파일(실행 로그, 테스트 결과 로그, 테스트 커버리지 로그, 메모리 누수 로그, 코드 분석 로그)을 지원한다.

4. 실험 및 평가

본 장에서는 제 3장에서 설계한 로봇 컴포넌트 테스트 자동화 프레임워크의 유효성과 효과성을 검증하기 위하여 그림과 같은 테스트 베드를 구축하였다.

4.1 실험 환경

테스트 엔진 서버는 Java 및 플렉스 언어를 이용하여 Linux 기반 PC에 구현하였고 로봇 소프트웨어 컴포넌트의 지속적인 통합을 위한 버전 관리 서버는 Collabnet사의 SVN 도구[6]를 이용하였다. 빌드/실행 에이전트는 C++ 언어를 이용하여 Linux 기반 로봇 환경에 구현하였고 로봇 컴포넌트 소스 및 테스트 자원 정보를 다운로드 하기위해 SVN 클라이언트를 설치하였다.

4.2 실험 결과 및 평가

본 논문에서는 RTC 기반 센서 컴포넌트의 기능적 클래스에 대한 단위 테스트를 수행한다. RTC 기반 센서 컴포넌트는 C++ 언어를 통해 구현되었으며 개발자는 SVN 클라이언트를 이용하여 버전 관리 서버에 변경된 내용이 있을 때마다 지속적으로 업로드 한다. 그림 4는 센서 컴포넌트의 기능적 클래스에 대해 테스트를 수행한 결과를 나타낸다.

본 논문에서 제안한 테스트 자동화 프레임워크는 SVN 도구를 이용한 버전 관리 서버를 통해 지속적으로 로봇 소프트웨어 컴포넌트의 소스를 통합하였고 경량 빌드/실행 에이전트를 이용하여 다양한 로봇 환경에서 해당 로봇 소프트웨어 컴포넌트가 동작하는지 검증 할 수 있었다. 또한 웹 사용자 인터페이스를 통해 여러 테스트 케이스 생성 방법을 이용하여 테스트 케이스 및 테스트 드라이버(C, C++, Java)를 자동 생성한 후 실행함으로써 사용자가 로봇 소프트웨어 컴포넌트를 손 쉽게 테스트 할 수 있게 하였다.

1	-93876.525163	-93876.525163	+	#-10000,10000	-187753.050826	FAIL.
2	-93876.525163	-93876.525163	/	#-10000,10000	1.000000	PASS.
3	-93876.525163	-93876.525163	-	#-10000,10000	-62584.350109	FAIL.
4	-93876.525163	-93876.525163	-	#-10000,10000	0.000000	PASS.
5	-93876.525163	-100.000000	+	#-10000,10000	-93976.525163	FAIL.
6	-93876.525163	-100.000000	/	#-10000,10000	539.765252	PASS.
7	-93876.525163	-100.000000	-	#-10000,10000	-31325.506380	FAIL.
8	-93876.525163	-100.000000	-	#-10000,10000	-93976.525163	FAIL.
9	-93876.525163	-76.018555	+	#-10000,10000	-93952.543718	FAIL.
10	-93876.525163	-76.018555	/	#-10000,10000	1234.915938	PASS.
11	-93876.525163	-76.018555	-	#-10000,10000	-31317.514573	FAIL.
12	-93876.525163	-76.018555	-	#-10000,10000	-93800.506608	FAIL.
13	-93876.525163	100.000000	+	#-10000,10000	-93976.525163	FAIL.
14	-93876.525163	100.000000	/	#-10000,10000	-938.765252	PASS.

<그림 4> 로봇 센서 컴포넌트 테스트 결과

5. 결 론

신뢰성이 결여된 로봇 소프트웨어의 사용은 인명피해 또는 경제적 손실과 직결되기 때문에 고품질의 로봇 소프트웨어 컴포넌트 개발은 중요한 연구과제이다.

본 논문에서는 위와 같은 고려사항을 만족하는 컴포넌트 기반 로봇 소프트웨어의 지속적인 통합 및 계층적 테스트 자동화를 위한 분산 프레임워크 구조를 제안하였다. 제안한 테스트 자동화 프레임워크의 유효성과 효과성을 입증하기 위하여 RTC 기반 로봇 센서 컴포넌트에 대해 테스트 시나리오에 따라 실험하였다.

결과적으로 테스트 자동화 프레임워크를 이용할 경우 지속적인 로봇 컴포넌트의 통합, 다양한 로봇 환경에서의 테스트, 다양한 테스트 케이스 및 드라이버 코드 생성이 가능하다는 것이 증명되었다.

[참 고 문 헌]

- [1] G.J.Myers. "The Art of Software Testing", John Will & Sons, 1979
- [2] <http://martinfowler.com/articles/continuous>
- [3] The Open Group, TETware White Paper, March, 2003
- [4] OASIS, ebXML Test Framework Committee Specificatin V1.0, March 2003
- [5] M.R.Blackburn, R.Busser and A.Nauman, "Eliminating Requirement Defects and Automating Test", Software productivity Consortium, 2001
- [6] <http://www.open.collab.net/downloads/subversion/>