
OPRoS 컴포넌트를 위한 구조적 적합성 시험 도구 설계 및 구현

Design and Implementation of the Structural Conformance Test Suite for OPRoS Component

이태희*, 김준*, 안대영*, 심정민*, 송병열**, 김성훈**, 정영숙**, 김주만***
(주)테스트마이다스*, 한국전자통신연구원**, 부산대학교 바이오정보전자공학과***

Tae-Hee Lee(thlee@testmidas.com)*, June Kim(jkim@testmidas.com)*,
Dae-Young Ahn(dyahn@testmidas.com)*, Jeong-Min Shim(jmshim@testmidas.com)*,
Byung-Yeol Song(sby@etri.re.kr)**, Sung-Hoon Kim(saint@etri.re.kr)**,
Young-Sook Jeong(ysjeong@etri.re.kr)**, Joo-Man Kim(joomkim@pusan.ac.kr)**

요약

최근 우리 생활에 로봇 시스템이 빠르게 확산되고 있으며, 로봇의 기능성에 대한 사용자의 요구가 증대됨으로써, 로봇 소프트웨어 컴포넌트는 세분화 되고 더욱 복잡하게 되었다. 따라서 표준화된 소프트웨어 플랫폼의 채택이 불가피하게 되었다. OPRoS는 로봇 시스템을 위한 컴포넌트 기반의 개방 표준 소프트웨어 플랫폼으로 소프트웨어 컴포넌트가 개발되면, OPRoS 표준 규격을 따르는 로봇 소프트웨어 컴포넌트의 적합성 검증이 필수적이다. 본 논문에서는 사용자들에 의해 개발된 각 컴포넌트의 적합성을 검증하는 구조적 적합성 테스트 도구에 대한 설계와 구현에 대하여 다룬다. 일부 실험적 테스트의 결과로서 컴포넌트가 표준 요구를 수용하는지, 수용하지 않는지를 본 테스트 도구로서 확인할 수 있었다.

■ 중심어 : | OPRoS | 컴포넌트 | 로봇 | 테스트 | 검증 |

Abstract

In recent years, there has been a rapid and wide spread of robot system in our lives and with the ever increasing the requirement of user for robot functionality, the software component is increasingly sophisticated and complicated. Accordingly it is inevitable to employ a standardized software platform. The OPRoS is open standard software platform based on component for robot system. When the software components were developed, it has got to validate the conformity of robot software components which comply with the OPRoS standard specification. In this paper, we deal with the design and implementation of the structural conformance test suite to verify and validate the conformance of components to be developed by user. As the results of some experiments, we could verify it from this test suite whether the components satisfied standard requirement or not.

■ keyword : | OPRoS | Component | Robot | Testing | Validation |

1. 서론

로봇 산업은 21세기 유비쿼터스 시대에 IT 및 BT 등의 융합 기술로서 새로운 지능형 로봇의 패러다임을 이

끌어갈 주요 산업으로 부각되고 있다. 지능형 로봇 시스템은 그 기술의 범위가 매우 넓으며, 메커니즘, 감지, 인지 및 감성 기술들을 포함한 자율행동, 상호 작용 등과 같은 지능적 기술들을 포함하며, 응용 분야 측면에

* 본 연구는 지식경제부와 한국산업기술평가관리원이 제공한 산업원천기술개발 사업인 지능형 로봇 테스트베드 운영 과제 의 연구비를 지원받아 작성되었다.[과제번호: 1034933].

접수번호 : #100330-004

접수일자 : 2010년 03월 30일

심사완료일 : 2010년 06월 15일

교신저자 : 김주만, e-mail : joomkim@pnu.ac.kr

서 다양한 서비스를 정의한 컴포넌트 모듈로 정의할 수 있다. 기존 로봇은 Stand-alone 형으로 소프트웨어 구조가 경직되었고, 소프트웨어 모듈 인터페이스가 모듈 간 상호 의존성에 의한 호환성 부족으로 이식성, 확장성 및 재사용성이 어려운 실정이며, 같은 기능의 로봇을 제조하면서 서로가 호환되지 않는 컴포넌트를 생산함으로써 국가적 경쟁력 저하는 물론 개발 기간과 많은 불필요한 비용이 소요되는 실정이다[6].

URC (Ubiquitous Robotic Companion)는 네트워크 기술을 기반으로 언제 어디서나 사용자와 함께 하며, 사용자에게 필요한 다양한 서비스를 제공하는 로봇의 표준 플랫폼으로 국가 신성장 동력 산업으로 지능형 로봇의 개념을 제시하였다[5]. 로봇 산업이 철도, 항공 그리고 자동차 등과 같이 부품기술과 통합기술의 필요성으로 기술 개발 기간의 단축과 제품 개발의 효율성을 높이기 위한 표준화가 요구되었다. 이에 국가는 표준화 정책의 일환으로 로봇SW플랫폼 기술개발의 표준화를 추진하게 되었다. 과거 정동부와 산자부에서 각각 RUPI(Robot Unified Platform Initiative)와 SPIRE(Software Platform Initiative for Robotics Engineering)라는 기술개발을 추진하였으며, 새 정부 들어 OPRoS라는 이름으로 표준 규격을 일원화 하였다[7].

OPRoS는 여러 종류의 로봇 시스템들의 응용 프로그램들을 동작 시킬 수 있는 개방형 SW 구조로서 컴포넌트의 종류에 따른 규격이 정해진 상태이며, S/W 프레임워크도 만들어진 상태이다[1][8]. 그러나 OPRoS 컴포넌트 개발도구의 사용사례가 전무하고 그에 대한 검증사례 또한 미흡한 실정이다.

OPRoS 로봇 컴포넌트는 표준화된 구성요소와 인터페이스로 구성되어 서로 다른 컴포넌트와 연동하여 로봇 응용이 실행되는 구조이다. 만약 작성된 컴포넌트가 OPRoS에서 정의한 컴포넌트 규격과 상이하거나 비정상적인 방법으로 작성되어 있다면 정상적인 로봇 응용이 어려우므로, 이를 방지하기 위해 로봇 소프트웨어 컴포넌트가 표준 규격을 따르는지에 대한 검증이 먼저 필요하다[4]. [2]에서 제안한 객체지향 방법의 컴포넌트 테스트 프레임웍은 컴포넌트에 정의된 각 기술자

(descriptor)들을 테스트 하고자 하는 컴포넌트로 가져와야하기에 다소 복잡하고 자동화에 걸림돌이 되었다. 그러나 본 논문에서는 OPRoS 표준 규격에 맞도록 스키마를 정의해 두고, 개발자의 프로파일과 생성된 소스 코드를 통하여 자동적으로 컴포넌트 검증 절차를 거치는 구조적 적합성 시험 도구를 설계하고 구현한 결과를 제시한다.

본 논문은 OPRoS 로봇 컴포넌트 규격에 대한 구조적 적합성 시험도구의 설계 및 구현에 관한 것으로 제 2장에서 OPRoS시스템에 대하여 소개한다. 제 3 장에서는 적합성 테스트를 위한 범위에 대하여 기술하고, 제 4장은 적합성 도구의 설계 및 구현에 대하여 기술하였고, 제 5 장은 구현된 적합성 도구를 사용하여 예제 컴포넌트에 의한 시험 검증 및 평가를 기술하였고, 제 6 장에서 결론을 맺었다.

II. OPROS 시스템

1. 시스템 개요

OPRoS 시스템은 로봇 제어에 필요한 소프트웨어 컴포넌트를 수용하여, 주어진 목적 작업을 수행할 수 있도록 설계된 하드웨어 디바이스들과 로봇의 기능을 구현한 소프트웨어 알고리즘이 효과적으로 통합 운용되게 하는 기반 소프트웨어 플랫폼이다[8].

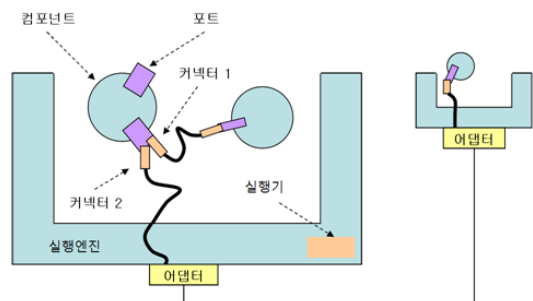


그림 1. OPRoS 시스템

OPRoS는 [그림 1]과 같이 실행 엔진과 실행 엔진 내부에서 실행되는 컴포넌트들로 구성된다. 실행 엔진은

컴포넌트들을 실행시키는 실행기를 가지며, 컴포넌트는 외부와의 통신을 위해서 포트를 가진다. 그리고 각 포트와 포트를 연결하기 위해서 커넥터가 필요하다. 또한 다른 실행 엔진에 있는 컴포넌트 사이의 통신을 위하여 어댑터가 지원된다.

2. 컴포넌트 모델

로봇 컴포넌트는 컴포넌트가 제공하는 인터페이스를 통해 접근되고 재사용 및 교체가 가능한 소프트웨어 모듈로서, 외부 컴포넌트 사용자 입장에서는 컴포넌트가 제공하는 인터페이스만을 이용하며 해당 인터페이스에 대한 상세 구현은 알 수 없게 된다. 즉, 컴포넌트는 외부에 공개된 인터페이스 이외에는 알 수 없도록 정보는닉하며, 이렇게 함으로서 외부 인터페이스가 변경되지 않는 한 내부 구현을 자유롭게 변경할 수 있다.

로봇에서 주로 사용되는 개발 패턴은 크게 두 가지로서, 첫째는 컴포넌트 사이에 주기적으로 교환되는 데이터를 처리하거나 로봇의 장치를 제어하는 방식이며, 둘째는 컴포넌트에서 제공하는 메소드(method)를 다른 컴포넌트에서 호출하여 컴포넌트를 제어하는 방식이다 [3]. OPRoS 컴포넌트는 상기 두 가지 형태의 로봇 컴포넌트 개발 방식을 모두 지원한다.

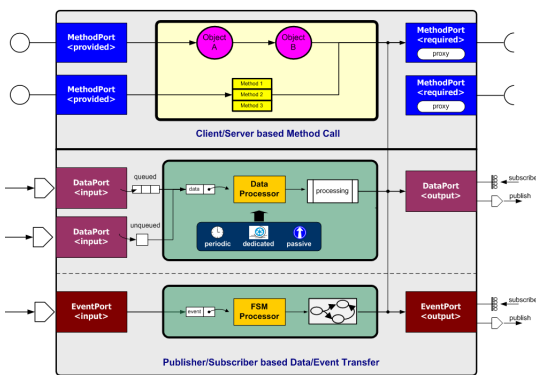


그림 2. 컴포넌트 모델

[그림 2]에서처럼 컴포넌트는 서비스 포트를 통해 다른 컴포넌트가 제공하는 메소드를 호출하거나 해당 컴포넌트의 속성에 접근할 수 있는 Client/Server 모델과

데이터/이벤트 포트를 통해 해당 컴포넌트에 데이터나 이벤트를 전달하는 Publisher/Subscriber 모델로 구성된다. 컴포넌트간의 메소드 호출이나 데이터/이벤트 전달은 포트를 통해 이루어지기 때문에, 다른 컴포넌트의 포트의 규격을 알아야 한다. 따라서 컴포넌트 개발자는 해당 컴포넌트가 제공하는 포트와 다른 컴포넌트가 사용할 수 있는 포트의 규격을 프로파일에 명시하여야 한다.

3. 실행기(Execution Engine)

컴포넌트가 자체의 알고리즘 수행에 충실할 수 있도록 실행기는 실행관련 부분을 따로 관리한다. 실행기는 실행 중에 컴포넌트에게 동작과 관련된 지시를 함수 호출 형식으로 하게 된다.

실행기를 두는 이유로는 각 컴포넌트가 자신의 쓰레드를 가지며 독립적으로 수행하면 부하가 너무 커지기 때문이다. 그룹으로 수행이 되거나, 수행의 방법이 비슷하다면(예, 같은 주기에 의한 실행) 동일한 실행기에 의해서 관리하는 것이 효율적이다.

4. 커넥터(Connector)

컴포넌트에서는 포트를 통해서 다른 컴포넌트의 서비스 호출을 할 수 있고, 데이터를 보내거나 받을 수 있다. 포트는 컴포넌트 개발자가 사용하는 인터페이스이고, 커넥터는 이러한 포트를 통한 커뮤니케이션을 가능하게 해주는 연결로 프레임워크 개발자가 구현한다.

5. 프로파일 스키마(Profile Schema)

OPRoS는 컴포넌트 프로파일, 컴포징 컴포넌트 프로파일, 애플리케이션 프로파일, 노드 프로파일, 메소드 프로파일, 데이터/이벤트 프로파일을 가지며, 이들 6개의 프로파일에 대한 스키마 구조를 정의하고 있다.

III. 적합성 시험(Conformance Test) 범위

1. 표준 시험의 구성

OPRoS 로봇 컴포넌트는 표준화된 구성요소와 인터

페이스로 구성되어 있어야 하며 구성된 인터페이스와 구성요소를 통해 서로 다른 컴포넌트와 연동하여 로봇 응용이 실행된다.

로봇 소프트웨어 플랫폼을 개발하여 완성하기 까지 [표 1]에서 보는바와 같이 3단계의 시험 단계를 거쳐야 한다.

표 1. 적합성 시험 단계

시험단계	시험 내용
구조적 적합성 (Structural Conformance)	· 컴포넌트 프로파일의 OPRoS 규격 준수 여부 검증 · 컴포넌트 프로파일과 소스 코드 간 일관성 검증 · 소스코드 내에 필수 요소 구현 여부 검증
동적 적합성 (Dynamic Conformance)	· 컴포넌트의 기본 동작 검증 · 컴포넌트의 인터페이스 검증
행동 적합성 (Behavioral Conformance)	· 컴포넌트의 기능 검증 · 컴포넌트 기능의 통합 검증 · 로봇 Action Scenario 검증

첫 번째 단계는 컴포넌트에 관련된 컴포넌트 프로파일과 포트 프로파일들에 대한 OPRoS규격 준수여부와 각 프로파일과 소스코드의 일관성 등으로 적합성에 대한 시험을 하며, 이 단계를 구조적 적합성 시험이라 명칭 한다. 이러한 구조적인 시험의 검증이 완료되고 나면 두 번째 단계로, 실행 엔진을 통한 컴포넌트의 동작/종료 등에 대한 기본 기능과 컴포넌트가 가진 포트들이 프로파일의 규격처럼 동작하는지에 대한 동적 적합성 시험을 수행한다. 마지막 세 번째 단계는 통합 검증 단계로 행동 적합성 검증을 시행하여 컴포넌트의 모든 기능 검증을 완료한다.

2. 표준 시험 범위

OPRoS 컴포넌트는 프로파일과 소스코드로 구성되며, 컴포넌트의 프로파일은 컴포넌트 자체 프로파일과 서비스 포트 프로파일, 데이터/이벤트 포트 프로파일로 구성된다. 각각의 프로파일들은 소스코드로 변환할 수 있도록 OPRoS에서 규격이 만들어져 있다. 이 규격에만 맞추어 프로파일을 작성하면 소스코드를 얻을 수 있고 해당 소스코드들로 컴포넌트를 만들 수 있다.

본 논문에서는 이들 프로파일과 생성된 소스코드를

대상으로 컴포넌트 구성에 관한 적합성 시험을 수행하는 구조적 적합성 시험 도구를 개발한다. 구조적 적합성 시험은 프로파일에 대한 완전성 검사와 구성 항목간의 일관성 검사 및 필수 구현 요소에 대한 검사로 단계적 실험이 이루어진다.

IV. 구조적 적합성 시험 도구 설계 및 구현

1. 설계 개요

OPRoS 로봇 컴포넌트에 대한 구조적 규격 적합성을 검증하는 시험도구는 4개의 패키지로 구성한다.

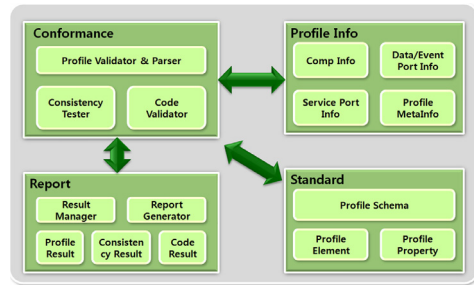


그림 3. 적합성 시험 도구의 구성 모듈

[그림 3]과 같이 시험 대상 컴포넌트에 대한 정보 관리를 위한 **Profile Info** 패키지, 적합성 시험 대상 항목들에 대한 정보 관리를 위한 **Standard** 패키지, 적합성 시험을 수행하는 **Conformance** 패키지 그리고 시험 결과를 관리하고 보고하는 **Report** 패키지로 구성하였다. 적합성 도구 구현을 위한 개발 환경으로 Java(JDK1.6.0)을 사용하였고, 개발 도구로는 Eclipse(3.4.1)를 사용하였다.

2. Conformance 패키지 설계

2.1 구조 및 기능

Conformance 패키지는 3 종류의 구조적 적합성 시험을 수행한다. [그림 4]와같이 **Profile Info** 모듈, **Standard** 모듈과 연동하여 시험을 수행하고, 수행한 시험 결과를 **Report** 모듈에게 전달한다.

- 시험 항목을 **Profile Info** 패키지 및 **Standard** 패키지에 요청
- 시험항목에 관련된 컴포넌트 정보들을 **Profile Info** 패키지에 요청
- 적합성 시험 수행(필수 항목 검사 및 항목 제한 개수 검사, 항목 간 일관성 검사, 소스 코드 필수 구현 요소 검사)
- 시험 결과를 **Report** 패키지에 전달

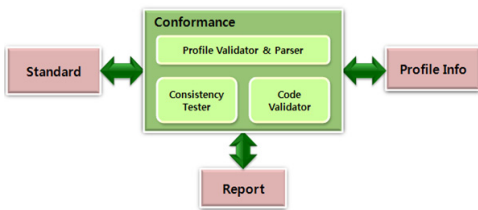


그림 4. Conformance 패키지 구조 및 외부 인터페이스

2.2 클래스 정의

[그림 5]에서처럼 Conformance 클래스를 메인 클래스로 하여 프로파일 검증, 일관성 검사 및 소스코드 필수 요소 검사를 수행할 수 있는 다수의 메소드를 포함한 클래스로 구성된다.

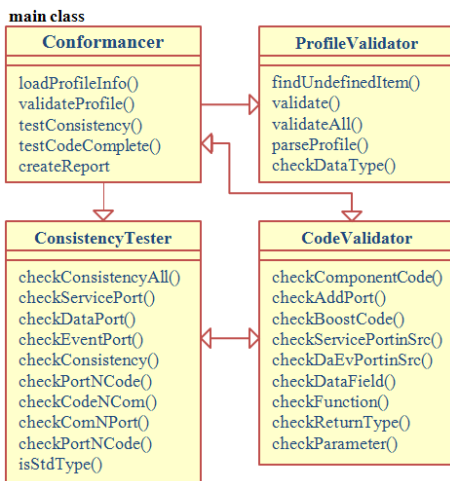


그림 5. Conformance 패키지 클래스 다이어그램

- Conformance Class
시험 도구의 main 클래스로 각각의 검사 클래스를 호출하여 시험 수행, 시험 결과 보고 생성
- ProfileValidator Class
기술문서의 완전성을 검사하는 클래스
- ConsistencyTester Class
모든 포트에 대하여 구성 항목 간 일관성 검사하는 클래스
- CodeValidator Class
소스코드에서 필수 구성요소를 검사하는 클래스로서 관련 기술문서에 해당하는 소스코드의 필수 구성요소와 소스코드간의 일관성 검사

3. Profile Info 패키지 설계

3.1 구조 및 기능

Profile Info 패키지는 적합성 시험의 대상이 되는 프로파일들의 세부 정보를 유지/관리한다. 컴포넌트, 서비스 포트, 데이터/이벤트 포트의 3가지 프로파일에 기술된 정보들 중 시험에 필요한 정보들을 분석하여 관리하고, [그림 6]에서 보는바와 같이 **Conformance** 패키지에서 시험을 수행할 때, 필요한 정보를 제공한다.

- 컴포넌트, 서비스 포트, 데이터/이벤트 포트 프로파일 및 메타 정보 유지/관리
- **Conformance** 패키지에게 컴포넌트 관련 프로파일 정보 전달.

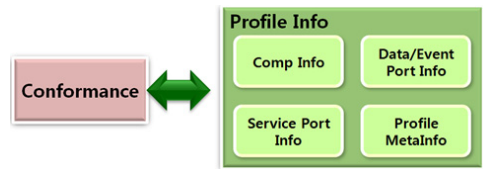


그림 6. Profile Info 패키지 구조 및 외부 인터페이스

3.2 클래스 정의

[그림 7]에서처럼 ProfileManager 클래스를 상위 클래스로 ComponentInfo, ServicePortInfo 및 DaEvPortInfo로 이루어진 클래스와 메타 정보를 관리

하는 클래스로 구성된다.

- ProfileManager Class

각각의 기술문서들의 정보를 유지 및 관리하는 클래스로서 기술문서 리스트를 얻어와 유효성 검사를 하고 Component Service Port, Data/Event Port에 기술된 정보들 중에서 시험에 필요한 정보들을 제공한다.

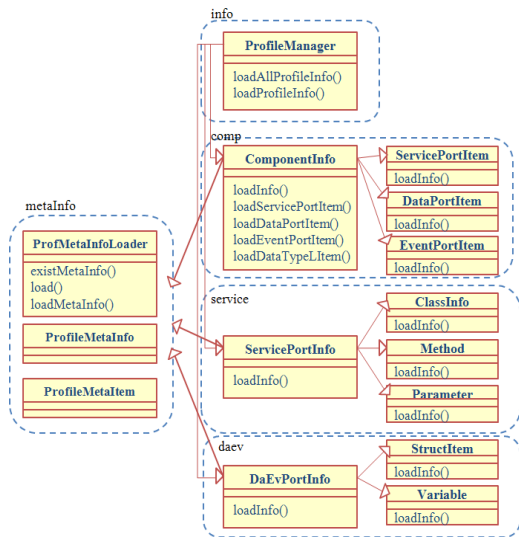


그림 7. Profile Info 패키지 클래스 다이어그램

- ComponentInfo Class

컴포넌트 기술문서의 정보를 관리하는 클래스로서 컴포넌트의 기본 정보와 서비스 포트, 데이터/이벤트 포트의 정보를 관리하며 메타정보와 컴포넌트 기술문서의 기본 정보를 비교한다.

- ServicePortItem Class

컴포넌트 기술문서내의 서비스 포트의 세부 항목 정보를 관리하는 클래스

- DataPortItem Class

컴포넌트 기술문서내의 데이터 포트의 세부 항목 정보를 관리하는 클래스

- EventPortItem Class

컴포넌트 기술문서내의 이벤트 포트의 세부 항목 정보를 관리하는 클래스

- ServicePortInfo Class

서비스 포트 기술문서의 정보를 관리하는 클래스

- ClassInfo Class

서비스 포트 클래스에 대한 정보를 관리한다.

- Method Class

서비스 포트 메소드에 대한 애트리뷰트 정보를 관리한다.

- Parameter Class

서비스 포트 메소드에 대한 전달인자 정보를 관리한다.

- DaEvPortInfo Class

데이터/이벤트포트 기술문서의 정보를 관리하는 클래스로서 struct 클래스 정보를 관리한다.

- StructItem Class

데이터/이벤트 포트 기술문서의 데이터 구조를 관리하는 클래스로서 각 데이터의 필드를 관리한다.

- Variable Class

데이터/이벤트 포트 기술문서의 데이터 구조내의 각각의 데이터 필드에 대한 클래스

- ProfileMetaInfoLoader Class

적합성 시험을 위해 기술문서에서 얻어야 하는 정보에 대한 메타정보를 로딩하는 클래스

- ProfileMetaInfo Class

기술문서에서 얻어야하는 항목들에 대한 메타정보를 관리하는 클래스

- ProfileMetaItem Class

포트 기술문서에서 얻어야하는 항목들에 대한 메타정보의 세부 항목을 관리하는 클래스

4. Standard 패키지 설계

4.1 구조 및 기능

Standard 패키지는 적합성 시험 대상 항목들에 대한 정보를 유지/관리한다. 적합성 시험 항목별로 세부 시험 항목들을 관리하고, **Conformance** 패키지에서 시험을 수행할 때, 필요한 정보를 제공하며, 시험 종료 후에는 **Report** 패키지에서 시험 결과와 항목들을 연결할 수 있도록 시험 항목을 전달한다. 그림 8은 이러한 과정을 도시하고 있다.

- 세부 시험 항목 관리
- 기술 문서 완전성 검사에서 수행할 세부 시험 항목 전달

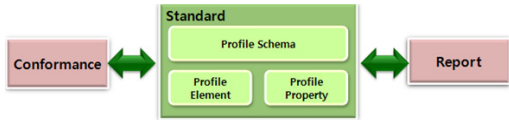


그림 8. Standard 패키지 구조 및 외부 인터페이스

4.2 클래스 정의

[그림 9]에서처럼 SchemaManager 클래스를 상위 클래스로 ProfileSchema, ProfileElement, ProfileReference 및 ProfileAttribute 클래스로 구성된다.

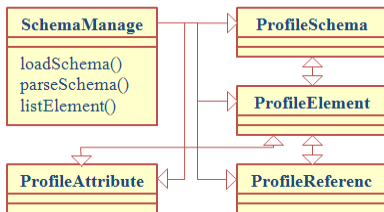


그림 9. Standard 패키지 클래스 다이어그램

- SchemaManager Class
모든 스키마 정보를 관리하는 클래스로서 기술문서 완전성 검사를 위한 스키마 정보를 제공
- ProfileAttribute Class
기술문서에 기술하여야 하는 애트리뷰트에 대한 규격 정보를 관리하는 클래스로서 기술문서 완전성 검사시에 사용된다. (함수생략)
- ProfileElement Class
기술문서에 기술하여야 하는 애트리뷰트에 대한 규격 정보를 관리하는 클래스로서 기술문서 완전성 검사시에 사용된다. (함수생략)
- ProfileReference Class
기술문서에 기술하여야 하는 애트리뷰트에 대한 규격 정보를 관리하는 클래스로서 기술문서 완전성 검사시에 사용된다. (함수생략)

- ProfileSchema Class
기술문서의 스키마 문서를 관리하는 클래스로서 기술문서 완전성 검사시에 기술문서의 스키마문서 정보를 제공한다.(함수생략)

5. Report 패키지 설계

5.1 구조 및 기능

Report 패키지는 적합성 시험 결과를 관리/보고한다. Conformance 패키지로부터 수행된 시험 결과를 전달 받아 각 시험 항목별로 세부 시험 항목에 대한 결과를 관리하며, 사용자에게 결과를 출력해준다. [그림 10]에서 Report 패키지와 Conformance 패키지간의 인터페이스를 도시하고 있다.

- Conformance 패키지로부터 시험 결과 전달 받음
- 시험 결과 정보 유지
- 시험 결과 문서 생성(Excel 문서로 생성)



그림 10. Report 패키지 구조 및 외부 인터페이스

5.2 클래스 정의

[그림 11]에서처럼 report 모듈의 ResultManager 클래스를 상위 클래스로 프로파일 검증 모듈, 일관성 검사 실험 및 소스코드 필수요소 검사 모듈로 구성된 각 클래스들로 구성되었다.

- ResultManager Class
모든 시험결과를 관리하는 클래스로서 각 시험에서 수행한 시험 결과를 전달 받는다.
- ReportGenerator Class
모든 시험결과를 수집하여 보고서를 생성하는 클래스로서 각 시험에서 수행한 시험 결과를 토대로 excel 형식의 보고서를 생성한다.

- **ResultSummary Class**
모든 시험결과를 요약 관리하는 클래스로서 완전성, 일관성, 필수 구현요소 검사 결과의 요약 정보를 제공한다.
- **ValidationResult Class**
기술문서 완전성 검사의 모든 결과를 관리하는 클래스(함수생략)
- **ProfileResult Class**
기술문서 완전성 검사의 모든 결과를 관리하는 클래스(함수생략)
- **TestItemInfo Class**
기술문서 완전성 검사의 각 항목에 대한 검사 결과를 관리하는 클래스(함수생략)
- **ValidationError Class**
기술문서 완전성 검사의 에러 정보를 관리하는 클래스(함수생략)
- **ConsistencyResult Class**
기술문서, 소스코드의 구성 항목 간 일관성 검사의 모든 결과를 관리하는 클래스(함수생략)
- **PortResult Class**
각 포트에 대한 일관성 검사의 결과를 관리하는 클래스(함수생략)
- **TargetResult Class**
각 포트에 대한 일관성 검사의 시험 대상에 대한 정보를 관리하는 클래스(함수생략)
- **ItemTestResult Class**
각 포트에 대한 일관성 검사의 시험 대상에 대한 시험결과 정보를 관리하는 클래스(함수생략)
- **RequiredCodeResult Class**
소스코드에 대한 필수 구현요소 검사 결과를 관리하는 클래스(함수생략)
- **CodeTarget Class**
소스코드에 대한 필수 구현요소의 각 검사 대상 정보를 관리하는 클래스(함수생략)
- **CodeItemResult Class**
소스코드에 대한 필수 구현요소의 각 검사 항목 및 결과를 관리하는 클래스(함수생략)

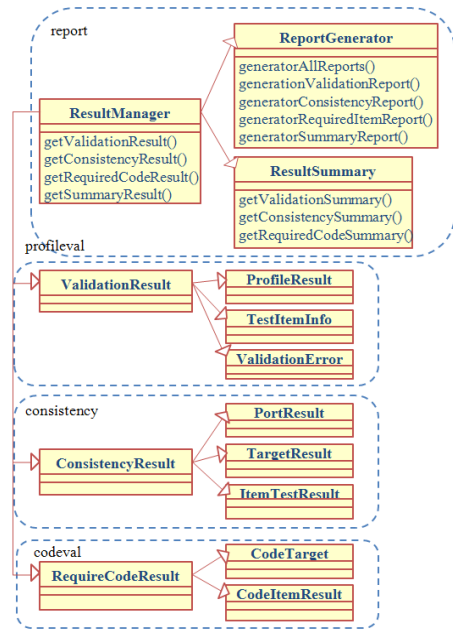


그림 11. Report 패키지 클래스 다이어그램

V. 시험 결과 및 검증

1. 시험 방법

OPRoS 로봇 컴포넌트 규격 적합성 시험 도구는 윈도 환경의 명령 프롬프트에서 실행 되도록 구성하였다. 개발된 도구에 다양한 에러 상황을 발생시키는 예제 파일을 이용하여 적합성 시험 도구를 검증한다. 예제 컴포넌트는 8개로써, 정상적인 컴포넌트와 오류를 포함하는 7개 컴포넌트로 구성하였다. 또한 동일한 방법으로 OPRoS 공식 사이트에 공개된 컴포넌트 8개를 시험하였다.

도구 실행은 프로파일의 경로, 소스파일의 경로 및 생성할 리포트 파일명 등 3개의 파라미터를 입력받아 실행한다. [그림 12]는 테스트 도구에 의한 실험 과정을 보여주고 있는데, example-test.bat 는 실행 스크립트 파일로서 내부에 이들 경로를 명시하고 있다.



그림 12. OPRoS 적합성 테스트 수행 화면

2. 시험 및 결과

예제 컴포넌트에 대한 적합성 시험을 수행하면 [그림 13]과 같이 엑셀형태의 결과파일 형태로 출력된다.

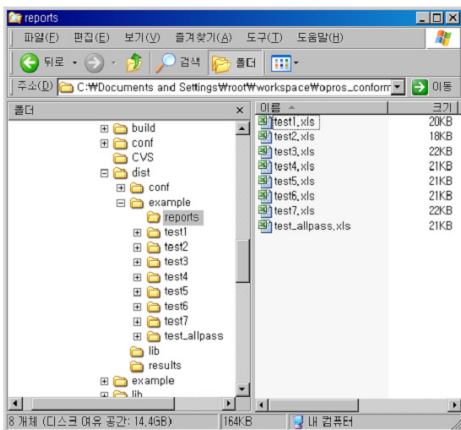


그림 13. 시험 결과 리포트 파일 생성

각 리포트 파일들은 엑셀형식으로 출력되며 각각의

시트별로 요약페이지, 기술문서 완전성 검사 페이지, 구성 항목 간 일관성 검사 페이지, 필수항목 구현여부 페이지, 등으로 구성된다.

[그림 14-17]은 위의 시험 결과 보고서파일의 내용들 중 하나의 예제이며 각각의 보고서 파일을 통하여 다양한 컴포넌트 에러에 대한 검증을 확인하였다.

[그림 14]는 적합성 시험 결과 요약 페이지로서 모든 시험의 결과에 대한 검사 항목 수와 PASS, FAIL, N/A에 대한 검사 결과 수를 한눈에 보여주는데 구성 항목 간 일관성 검사에서 12개의 FAIL이 있고 나머지 항목에서는 모두 PASS되었음을 보여준다.

	A	B	C	D	E	F	G
1							
2							
3		(ColorRecognitionComponent) 적합성 시험 결과 요약					
4							
5		시험 항목	검사 항목 수	PASS	FAIL	N/A	
6		기술문서 완전성 검사	7	7	0	0	
7		구성 항목간 일관성 검사	14	2	12	0	
8		필수 구현 요소 검사	6	6	0	0	
9							
10							

그림 14. 적합성 시험 결과 요약 페이지

[그림 15]는 기술문서 완전성 검사 페이지로서 컴포넌트 프로파일, 서비스 포트 프로파일, 데이터/이벤트 포트 프로파일에 대한 OPRoS 규격 준수 여부를 확인할 수 있다.

	A	B	C	D	E		
1							
2							
3		기술문서 완전성 검사					
4							
5		기술 문서 분류	기술 문서명	검사 대상	시험 결과(P/F)		
6		component	ColorRecognitionComponent.xml	ColorRecognitionComponent : 기본 정보	PASS		
7				ColorRecognitionComponent : 실행 환경	PASS		
8				ColorRecognitionComponent : 실행 semantics	PASS		
9				ColorRecognitionServicePort : 요약 정보	PASS		
10				CameraServicePort : 요약 정보	PASS		
11				ImageData : 요약 정보	PASS		
12				service	ColorRecognitionServicePort.xml	ObjectRecognitionServicePort : 테스트 정의	PASS
13							
14							

그림 15. 기술문서 완전성 검사 페이지

[그림 16]은 구성 항목 간 일관성 검사 페이지로서 각각의 포트에 대한 컴포넌트 프로파일과의 일관성, 소스 코드에 대한 일관성, 해당 컴포넌트 프로파일과 소스코드와의 일관성에 대한 검사 결과를 확인할 수 있다.

[그림 17]은 소스코드 필수 구현 요소 검사 페이지로

서 컴포넌트와 각 포트의 소스코드에 대한 필수 구현 요소를 검사한 결과를 확인할 수 있다.

시행 항목	포트 명	검사 대상	대상 항목	검사 항목	시험 결과(P / F / N/A)
ColorRecogni- onServicePort	필수 소스 코드 vs. 포트 기 용론서	ColorRecognitionComponent .im vs. ColorRecognitionServicePort. im	클래스(Class)명 일치 여부	PASS	
			클래스(Class)명 일치 여부	FAIL: 클래스(Class)명 불일치 (Comp. Profile: ColorRecognitionServicePort, Port Profile: ColorRecognitionServicePort)	
			클래스(Class)명 일치 여부	FAIL: 클래스(Class)명 불일치 (Port Profile: ColorRecognitionServicePort, Source Code: X)	
			클래스(Class)명 일치 여부	FAIL: 클래스(Class)명 불일치 (Port Profile: ColorRecognitionServicePort, Source Code: X)	
			클래스(Class)명 일치 여부	FAIL: 클래스(Class)명 불일치 (Port Profile: ColorRecognitionServicePort, Source Code: X)	
			클래스(Class)명 일치 여부	FAIL: 클래스(Class)명 불일치 (Port Profile: ColorRecognitionServicePort, Source Code: X)	
	소스 코드 vs. 필 수 소스 코드	ColorRecognitionComponentPro- file.im vs. ColorRecognitionServicePortPro- file.im	클래스(Class)명 일치 여부	PASS	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
필수 소스 코드 vs. 포트 기 용론서	ColorRecognitionComponent. im vs. unknown profile	클래스(Class)명 일치 여부	FAIL: port profile) 없음		
		클래스(Class)명 일치 여부	FAIL: port profile) 없음		
		클래스(Class)명 일치 여부	FAIL: port profile) 없음		
		클래스(Class)명 일치 여부	FAIL: port profile) 없음		
		클래스(Class)명 일치 여부	FAIL: port profile) 없음		
		클래스(Class)명 일치 여부	FAIL: port profile) 없음		
CameraService- Port	필수 소스 코드 vs. 포트 기 용론서	unknown profile vs. unknown source code	클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
			클래스(Class)명 일치 여부	FAIL: port profile) 없음	
소스 코드 vs. 필 수 소스 코드	unknown source code vs. ColorRecognitionComponent.im	클래스(Class)명 일치 여부	FAIL: source file를 찾을 수 없음		
		클래스(Class)명 일치 여부	FAIL: source file를 찾을 수 없음		

그림 16. 구성 항목 간 일관성 검사 페이지

위 결과들과 같이 적합성 시험 도구를 통해서 정상적인 컴포넌트와 오류가 있는 컴포넌트에 대한 검증결과를 확인하였고 사용자는 파라미터 정보를 가지고 도구를 쉽게 수행하여 검증할 수 있도록 했으며, 작업의 결과를 엑셀파일로 출력해 줌으로써 시험 결과에 대한 내용을 쉽게 알아볼 수 있게 하였다.

시행 항목	검사 대상	검사 내용	시험 결과(P/F)
component	ColorRecognitionComponent	Component 하위 Class 구현 여부	PASS
		getComponent() 구현 여부	PASS
		releaseComponent() 구현 여부	PASS
service_port	ColorRecognitionServicePort	addPort() 구현 여부	PASS
data_event_port	CameraServicePort	addPort() 구현 여부	PASS
		imageData	addPort() 구현 여부

그림 17. 소스코드 필수 구현 요소 검사 페이지

VI. 결론

로봇 개발자들에 의해 만들어지는 로봇 컴포넌트의 이식성과 효율성을 제고하기 위하여 개방형 로봇 S/W 표준 플랫폼을 지향하는 OPRoS 표준이 제정되었다. 따라서 모든 개발자들은 OPRoS 규격에 따라 각 컴포넌트를 개발하지만, 방대한 소프트웨어 및 문서들의 표준 준수 여부를 검증할 수 있는 도구가 부재한 실정 이었

다. 본 논문에서는 OPRoS 컴포넌트 표준 규격의 준수 여부를 검증하기 위하여 3단계 적합성 시험인 구조적 적합성, 동적 적합성 및 행동 적합성을 제시하였고, 이들 3 단계 적합성 시험 중 구조적 적합성 시험 도구의 설계 및 구현기술을 제안하였다.

컴포넌트 적합성 도구는 프로파일 완전성 검사, 프로파일과 소스코드 구성 항목 간 일관성 검사, 소스코드 필수 구현항목 검사의 3가지 기능 단위로 분류하였고, 각각의 시험 결과를 결과 리포트를 통해서 쉽게 확인할 수 있도록 하였다. 개발된 적합성 도구는 다양한 예제 프로파일 및 OPRoS 공식 사이트에 공개된 컴포넌트에 대한 실험 결과를 통하여 정확하게 검증할 수 있었음을 확인할 수 있었다.

본 논문을 통하여 제시한 구조적 적합성 검증은 로봇 컴포넌트 규격 검증이며, 실제 로봇의 기능이나 컴포넌트들 간의 연동 및 동작 상태에 대한 검증은 동적 적합성과 행동 적합성 도구에 의해 검증 될 수 있을 것이다. 따라서 본 연구의 결과를 바탕으로 동적 적합성 도구 및 행동 적합성 도구에 대한 연구가 진행되어야 할 것이다.

참고 문헌

- [1] B. Song, S. Jung, C. Jang, and S. Kim, "An Introduction to Robot Component Model for OPRoS," Intl. Conf. on SIMPAR 2008, 2008(11).
- [2] F. Jabeen and M. Jaffar-ur-Rehman, "A Framework for Object Oriented Component Testing," International Conference on Emerging Technologies, 2005(9).
- [3] H. Ahn, "Component Design Patterns for Standard Robot Component Platform," ICCAS, 2009(3).
- [4] S. Maeng and H. Park, "State Test Methodology for Robot Software Component Architecture," The 6th Inter. Conf. on Ubiquitous Robots and Ambient Intelligence'09, 2009.

- [5] 임성호, 김주만, “웹서비스 기반 URC 로봇 원격 모니터링 기술의 설계 및 구현”, 한국콘텐츠학회 논문지, 6권 11호, pp.285-294, 2006(11).
- [6] 정연규, 조현규, “지능형 로봇의 국제 표준화 동향”, 전자통신동향분석, 제22권, 제2호, 2007.
- [7] 신은철, 손수경, 박현섭, “로봇 S/W를 위한 OPRoS 컴포넌트”, 대한전자공학회 추계학술대회, CFP-1, 2008.
- [9] <http://www.opros.or.kr>

저 자 소 개

이 태 희(Tae-Hee Lee)

정회원



- 2005년 2월 : 밀양대학교 정보통신공학과(공학사)
- 2010년 2월 : 부산대학교 바이오정보전자공학과(공학석사)
- 2008년 7월 ~ 현재 : (주)테스트 마이다스 연구원

<관심분야> : 소프트웨어 테스트, 임베디드 시스템, 실시간 원격 제어, 웹 응용

김 준(June Kim)

정회원



- 1983년 : 부산대학교 계산통계학과(공학사)
- 1986년 : 한국과학기술원 전산학과(공학석사)
- 1986년 ~ 현재 : ETRI 저장시스템 연구팀 팀장

▪ 2008년 ~ 현재 : (주)테스트 마이다스 대표이사
 <관심분야> : 파일 시스템, 트랜잭션 처리, 소프트웨어 테스트

안 대 영(Dae-Young Ahn)

정회원



- 1984년 : 한양대학교 전자공학과(공학사)
- 1986년 : 한국과학기술원 전기 및 전자공학과(공학석사)
- 1999년 : 한국과학기술원 전기 및 전자공학과(공학박사)

▪ 1986년 ~ 2002년 : ETRI 팀장
 ▪ 2002년 ~ 2004년 : 서원대학교 컴퓨터교육과 교수
 ▪ 2005년 ~ 2007년 : 육성전자 연구소 그룹장
 ▪ 2008년 ~ 현재 : (주)테스트 마이다스 부사장
 <관심분야> : 시스템 소프트웨어, 실시간 운영체제, 소프트웨어 테스트

심 정 민(Jeong-Min Shim)

정회원



- 2002년 : 충북대학교 정보통신공학(공학사)
- 2004년 : 충북대학교 정보통신공학(공학석사)
- 2004년 3월 ~ 2006년 12월 : ETRI 연구원

▪ 2007년 1월 ~ 2008년 6월 : (주)모두택 선임연구원
 ▪ 2008년 7월 ~ 현재 : (주)테스트마이다스 선임연구원
 <관심분야> : 임베디드 시스템, 임베디드 시스템 테스트 자동화

송 병 열(Byung-Yeol Song)

정회원



- 1994년 : 전북대학교 전자공학과(공학사)
- 1996년 : 전북대학교 전자공학과(공학석사)
- 1997년 ~ 현재 : ETRI 선임연구원

<관심분야> : 임베디드 시스템, 실시간 운용체제, 로보제어소프트웨어

김 성 훈(Sung-Hoon Kim)

정회원



- 1995년 : 광운대학교 전자공학과(공학사)
- 1997년 : 광운대학교 전자공학과(공학 석사)
- 2007년 : 한양대학교 전자컴퓨터공학(박사수료)
- 1996년 ~ 현재 : ETRI 선임연구원
- 1998년 ~ 현재 : ETRI 지능로봇제어연구팀 팀장
<관심분야> : 로봇 소프트웨어 아키텍처, 컴포넌트 기반 로봇 응용 개발, 지능형 서비스 로봇

정 영 숙(Young-Sook Jeong)

정회원



- 1988년 : 이화여자대학교 전자계산학(공학사)
- 2001년 : 충남대학교 전자공학과(공학석사)
- 2005년 : 충남대학교 전자공학과(박사수료)
- 1989년 2월 ~ 현재 : ETRI 책임연구원
<관심분야> : 지능형 로봇 HW/SW 시험평가 및 표준화, Human Robot Interaction, 로봇통신응용

김 주 만(Joo-Man Kim)

정회원



- 1984년 : 숭실대학교 전자계산학(공학사)
- 2003년 : 충남대학교 컴퓨터공학과(공학박사)
- 1985년 1월 ~ 2000년 2월 : ETRI 책임연구원(OS팀장)
- 1995년 7월 ~ 1996년 6월 : Novell Inc. Research Center 방문 연구원
- 2000년 3월 ~ 2006년 2월 : 밀양대학교 정보통신공학부 교수
- 2006년 3월 ~ 현재 : 부산대학교 바이오정보전자 교수
<관심분야> : 임베디드 시스템, 실시간 운영체제, 웹 응용 소프트웨어, 로봇제어소프트웨어