

# 테스팅을 위한 컴포넌트 요소 식별

김지혁\*, 김종배\*\*, 류성열\*

\*승실대학교 컴퓨터학부

\*\* (주)이엔터프라이즈

e-mail:ji-hyeok@selab.ssu.ac.kr

## A Component Element Identification for Effective Testing

Ji-Hyeok Kim\*, Jong-Bae Kim\*\*, Sung-Yul Rhew\*

\*Dept. of Computing, Soongsil University

\*\*E-Enterprise Co.,Ltd

### 요 약

일반적으로 컴포넌트 기반의 SW는 바이너리 파일로 구성되어 있다. 즉, 블랙박스 형식이다. 따라서 컴포넌트 테스트 시에 코드기반의 화이트박스 테스트 기법을 적용하기가 어렵다. 그래서 컴포넌트 테스트 시에는 소스코드기반이 아닌 명세기반의 테스트를 수행해야 한다. 그러나 컴포넌트 제공자는 고객에게 바이너리 파일과 일부분의 명세만을 제공한다. 즉, 컴포넌트 제공자는 컴포넌트 안에서의 테스트 범주와 테스트 매트릭에 대한 정보를 제공하지 않는다. 이에 따라 고객은 컴포넌트의 테스트 적절성, 테스트 커버리지, 그리고 테스트 품질을 이해하는 것이 어렵다. 그리고 효과적인 테스트를 위해서는 적절한 테스트 모델이 있어야 한다. 따라서 본 논문에서는 컴포넌트 테스트를 효율적으로 하기 위하여 컴포넌트 테스트 모델을 제안하였고, 테스트를 위한 명세 요소를 식별하였다.

**Keyword** CBS, 테스트, 명세

### 1. 서론

컴포넌트는 in-house 컴포넌트, COTS 컴포넌트로 구분한다. in-house 컴포넌트는 재사용이나 업데이트의 목적으로 특정 도메인에 한하여 자체 개발한 컴포넌트이다. COTS 컴포넌트는 상용 컴포넌트를 말한다. 일반적으로 컴포넌트 기반의 SW는 COTS 컴포넌트를 지칭한다.

COTS 컴포넌트는 블랙박스 형태의 바이너리 파일로 제공된다. 따라서 컴포넌트 테스트 시에 코드기반의 화이트박스 테스트 기법을 적용하기가 어렵다. 따라서 컴포넌트 테스트 시에는 소스코드기반이 아닌 명세기반의 테스트를 수행해야 한다. 그러나 컴포넌트 제공자는 고객에게 바이너리 파일과 인터페이스 명세, 컴포넌트 명세만을 제공한다. 즉, 컴포넌트 제공자는 컴포넌트 안에서의 테스트 범주와 테스트 매트릭에 대한 정보를 제공하지 않는다. 이에 따라 고객은 컴포넌트의 테스트가 적절하게 이루어졌는지, 테스트 범위가 무엇인지, 테스트 품질이 어떠한지를 이해하기가 어렵다.

본 논문에서는 컴포넌트 테스트를 이해하고 원활한 테스트를 위하여 컴포넌트 테스트에 기반한 명세 요소를 식별하여 명세에 포함하고자 한다.

2장에서는 기존의 컴포넌트 명세의 요소가 무엇이 있는지, 그리고 CBS 테스트를 위한 범주를 구분하고, 요소를 식별한다. 4장에서는 결론을 맺고, 5장에서는 향후 본 연구를 기반으로 앞으로 나아가 연구 방향을 제시한다.

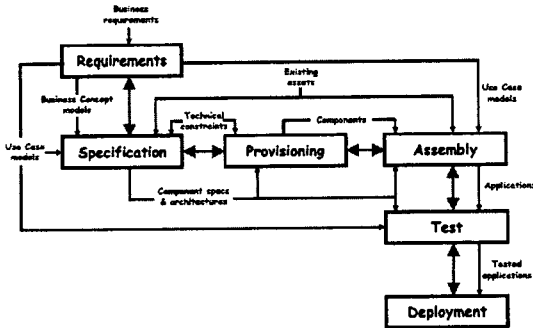
### 2. 관련연구

이 장에서는 기존의 컴포넌트를 명세화하는 산출물에는 어떤 것이 있는지, 그리고 CBS 테스트 시에 어떠한 어려움이 있는지에 대해 논의할 것이다.

#### 2.1 컴포넌트 명세

Cheesman이 제안한 UML Component[4]에서는 전체 개발 프로세스의 워크플로우에서 사용하는 산출물을 보여준다. 아래의 [그림 1]에서 설명된다. 요구 사항 정의와 명세 워크플로우에서 얻을 수 있는 주

요 산출물은 비즈니스 개념 모델과 유즈케이스 모델이다. 명세 워크플로우에서는 비즈니스 타입 모델, 인터페이스 명세, 컴포넌트 명세, 컴포넌트 아키텍처를 산출물로 얻을 수 있다.



[그림 1] 컴포넌트 개발 프로세스의 워크플로우  
본 논문에서는 in-house 컴포넌트가 아닌 COTS 컴포넌트를 대상으로 하기 때문에 명세 워크플로우에서의 인터페이스 명세와 컴포넌트 명세만을 컴포넌트 제공자가 제공할 것이다. 다음의 <표 1>은 인터페이스 명세와 컴포넌트 명세의 구성 요소이다.

<표 1> 명세에 따른 구성요소

명세명	설명	구성요소
인터페이스 명세	컴포넌트 오브젝트 사용자와의 계약	- 오퍼레이션의 시그니처 - 컴포넌트 오브젝트의 상태 - 오퍼레이션의 매개변수에 미치는 영향 - 어떠한 조건하에서 발생하는지 등에 관한 상세한 계약
컴포넌트 명세	인터페이스 명세와 제약 조건으로 정의	- 컴포넌트가 제공해야 할 인터페이스들용 정의 - 인터페이스 명세들이 어떻게 서로 간에 대응하는지 정의 - 사용하거나 소비하고 있는 인터페이스도 기술

## 2.2 CBS 테스트

CBS 테스트는 여타 다른 프로그램의 테스트와는 다르다. CBS 테스트는 소스코드기반의 화이트박스 테스트를 하기에는 어렵다. 따라서 제한된 정보를 바탕으로 CBS 테스트를 해야 한다. 즉, 제한된 정보인 인터페이스 명세나 컴포넌트 명세를 가지고 명세기반의 테스트를 해야 한다.

CBS 테스트가 어려운 이유는 Jerry Gao[5]가 다음과 같이 분류하였다.

(1) 소프트웨어 컴포넌트를 위한 테스트 스위트가 빈약함. 테스트 스위트(Test Suite)란 일정한 순서에 의하여 수행될 개별 테스트들의 집합, 또는 패키지를 말한다. 스위트는 응용분야나 우선순위, 내용에 연관된다.

(2) 컴포넌트 테스트의 적절성을 판단하기 힘들. 다시 말해서 컴포넌트 제공자가 컴포넌트 안의 테스트 범주와 테스트 매트릭에 대한 정보를 제공하지 않기 때문에 고객이 컴포넌트가 적절하게 테스트 되었는지, 테스트 범주가 무엇인지, 테스트 품질을 판단하고 이해하는 것이 매우 어렵다.

(3) 컴포넌트 테스트 드라이버나 스티브를 구성하는 가격이 매우 비쌈. 컴포넌트와 테스트 스위트, 그리고 테스트 베드를 가진 컴포넌트 사이의 잘 정의된 표준 인터페이스가 없기 때문이다.

따라서 CBS 테스트의 테스트성을 높이기 위해 다음 <표 2>와 같은 성질들이 추가되어야 한다.

<표 2> 테스트성을 높이기 위한 추가 성질

성질	설명
컴포넌트 식별성과 추적성	컴포넌트를 식별하고 컴포넌트안의 흐름을 추적할 수 있어야 한다.
컴포넌트 제어성	컴포넌트에 값을 입력하였을 때에, 입력값과 예상되는 출력값을 제어할 수 있어야 한다.

## 3. 컴포넌트 명세 요소 식별과 컴포넌트 모델

이 장에서는 2장에서 관련연구를 기본으로 컴포넌트 명세에 들어가 요소들을 식별한다. 또한 CBS 테스트 관점에서 명세에 들어가야 할 요소를 식별한다.

### 3.1 컴포넌트 명세 요소

컴포넌트 제공자가 고객에게 제공하는 명세는 인터페이스 명세와 컴포넌트 명세 이 두 가지이다. 본 연구에서는 컴포넌트 명세에 초점을 두어 명세 요소를 식별한다.

컴포넌트 명세에 대한 표준화 방안은 없다. 따라서 [3], [4], 마르미 III에서 다루는 컴포넌트 명세에서 식별할 수 있다. 따라서 다음과 같은 컴포넌트 명세의 요소들을 식별하였다.

명세명	구성요소
컴포넌트 명세	컴포넌트 이름
	서브시스템 이름
	컴포넌트가 제공해야 할 인터페이스
	컴포넌트가 사용하는 인터페이스
	컴포넌트 상호작용 제약사항
	인터페이스간의 관계
	컴포넌트 다이어그램
	컴포넌트 워크플로우 다이어그램
	커스터마이제이션 스키마

### 3.2 CBS 테스트 요소 식별

CBS의 테스트의 테스트성을 높이기 위해서는 컴포넌트 식별성과 추적성, 컴포넌트 제어성 등의 추가 성질들이 컴포넌트 명세안에 추가되어야 한다.

먼저, 컴포넌트의 식별성과 추적성을 높이기 위한 방법은 무엇이 있는가?

이것에 대한 해답은 컴포넌트 내부, 외부의 흐름을 알면 된다. 따라서 컴포넌트 내부 오퍼레이션, 내부 오브젝트 상태, 이벤트, 성능 등을 추적할 수 있어야 한다. 또한 컴포넌트 외부 오퍼레이션과 성능, 외부의 식별 가능한 이벤트, 그리고 공통 속성의 상태를 알고 있어야 한다.

따라서 다음 6가지의 타입의 trace을 알고 있어야 한다.

타입	설명
Operational trace	운용상의 자취를 알고 있어야 한다.
Performance trace	성능의 자취를 알고 있어야 한다.
Error trace	에러가 난 부분의 자취를 알고 있어야 한다.
State trace	오브젝트의 상태의 변화의 자취를 알고 있어야 한다.
Event trace	이벤트의 자취를 알고 있어야 한다.
Communication trace	상호교환의 자취를 알고 있어야 한다.

### 3.3 명세기반의 컴포넌트 테스트 모델

컴포넌트 명세만을 가지고 컴포넌트를 테스트해야 하기 때문에 명세기반의 컴포넌트 테스트 모델을 세워야 한다. 따라서 컴포넌트 명세에 있어서 큰 비중을 차지하는 컴포넌트 인터페이스와의 관계를 구분하고 요소를 식별하는 것 또한 중요하다.

다음은 인터페이스의 구분이다.

required interface : 컴포넌트의 가변성을 설정한다. 가변성이란 사용자의 설정에 의해서 컴포넌트 내부 구성이나 워크플로우가 변하는 것을 의미한다.
provide interface : 컴포넌트가 클라이언트에게 서비스를 제공하는 역할을 한다.
uses interdice : 서비스를 제공하는 컴포넌트로부터 필요한 서비스를 요청하는 역할을 한다.
platform interface : 다양한 플랫폼에 컴포넌트가 실행될 수 있도록 연결시켜주는 인터페이스이다.

이와 같은 인터페이스간의 상호교환의 특징을 모델화하기 위해서는 다음의 4가지 요소가 필요하다.

Interfaces

Events

Context-dependence Relationships

- 단일 컴포넌트에서의 상호작용이 일어날 때의 인터페이스와 이벤트

Content-dependence Relationships

- 다수의 컴포넌트에서의 상호작용이 일어날 때의 인터페이스와 이벤트

이 테스트 모델을 이용함으로써 명세만을 가지고 컴포넌트의 상호작용을 표현하는데 도움을 줄 것이다.

### 4. 결론

본 논문에서는 CBS 테스트의 어려움을 출발로 삼았다.

첫째, 소스코드기반의 화이트박스 테스트를 적용시키기가 어려워 명세기반의 테스트를 해야 한다.

둘째, 컴포넌트 제공자는 바이너리 파일과 명세 일부만을 제공한다. 따라서 CBS 테스트를 위한 범주, 테스트 커버리지, 테스트 매트릭 등을 제공하지 않는다. 따라서 CBS 테스트에 대해 고객이 이해하기 어렵다.

그러므로 CBS 테스트에 기반한 요소들을 식별하여, 컴포넌트 테스트 모델을 제안하였고, CBS 테스트를 위한 컴포넌트 명세의 요소를 도출하였다. 이렇게 함으로써 고객의 이해를 돕고 원활한 테스트를 수행할 수 있다.

### 5. 향후연구

향후 제안된 요소들을 기반으로 어떻게 컴포넌트 명세에 표현할 지 명세화시키는 방안을 연구할 것이다. 또한 제안된 컴포넌트 명세 요소들을 어떻게 표현해야 하는지에 대한 연구가 더 필요하다. 향후 컴

포넌트 명세화를 이루면 많은 CBS 개발자들이 염두에 두고 컴포넌트 명세를 작성할 수 있을 것이다. 후에 컴포넌트 테스트의 자동화를 위하여 컴포넌트 명세의 XML로의 변환을 추가 연구할 것이다.

#### 참고문헌

- [1] Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 5th Ed., McGraw-Hill, 2000
- [2] Jerry Gao, and Youjin Zhu, "Tracking Software Components", Technical report (<http://www.engr.sjsu.edu/gaojerry/techreport/>) in San Jose State University, in 1999. Submitted for publication.
- [3] George T. Heineman and William T. Councill, *Component-Based Software Engineering: Putting the Pieces Together*, Addison-Wesley, 2001
- [4] Cheesman, *UML Components: A Simple Process for Specifying Component-Based Software*, Addison-Wesley, 2001
- [5] Jerry Gao, "Testing Component-Based Software", STARWEST '99
- [6] Ye Wu, Mei-Hwa Chen, Jeff Offutt, "UML-based Integration Testing for Component-based Software", Second International Conference, ICCBSS 2003 Ottawa, Canada, 2003.
- [7] Elaine J Weyuker, "Testing Component-Based Software: A Cautionary Tale", *IEEE Software*, September/October 1998
- [8] Roy S. Freedman, "Testability of Software Components", *IEEE Transactions on Software Engineering*, Vol. 17, No. 6, June 1991.