

샤모아 프레임 워크의 컴포넌트 맞춤 테스트

이병희* 최병주
 이화여자대학교 컴퓨터학과
 {radia79, bjchoi}@mm.ewha.ac.kr

The Component Customization Test in the Chamois Framework

Byunghye Lee* Byoungju Choi
 Dept. of Computer Science & Engineering, Ewha Womans University

요 약

컴포넌트 기반 구조의 프레임 워크인 샤모아 프레임 워크(Chamois Framework)는 다양한 컴포넌트 아키텍처중 COM(Component Object Model)과 Web Service로 개발되었다. 컴포넌트 기반 소프트웨어 개발(Component-Based Software Development)은 이미 작성된 소프트웨어 컴포넌트를 자신의 목적에 맞도록 재사용함으로써 여러 가지 위험 요소가 따를 수 있다. 따라서, 샤모아 프레임 워크에서 맞춤으로 변형된 컴포넌트에서 발생하는 오류를 테스트 하기 위한 컴포넌트 맞춤 테스트 기법이 필요하다. 본 논문에서는 컴포넌트 맞춤 테스트 기법을 오류삽입 기반의 COM으로 확장하여 COM 컴포넌트 맞춤 테스트 기법을 샤모아 프레임 워크에 적용한 사례를 보인다.

1. 서론

컴포넌트 기반 소프트웨어 개발(Component-Based Software Development)은 컴포넌트를 재사용함으로써 CBSD에서의 테스트 중요성이 커지고 있다. CBSD는 이미 작성된 소프트웨어 컴포넌트를 재사용하여 소프트웨어 설계와 개발 비용감소를 가져올 수 있으나, 컴포넌트 사용자들이 자신의 목적에 맞도록 재사용함으로써 여러 가지 위험요소가 따를 수 있다. CBSD는 컴포넌트 맞춤(Component Customization) 테스트, 컴포넌트 조립(Component Composition) 테스트, 컴포넌트 시스템(Component System) 테스트, 컴포넌트 자격부여(Component Qualification) 테스트 등의 다양한 수준의 테스트를 요구한다.[1] 이 가운데 본 논문은 샤모아 프레임 워크 내에 컴포넌트를 개발 도메인의 특정 요구사항에 맞춤 때 요구되는 '컴포넌트 맞춤 테스트'를 대상으로 한다.

논문[2,3]에서는 공개하지 않는 컴포넌트의 핵심기능과 공개하는 컴포넌트의 인터페이스 부분을 일반적으로 표현하기 위하여 각각 블랙박스 클래스와 화이트박스 클래스로 정의하고, 컴포넌트 맞춤을 통해 변형된 화이트박스 클래스에 오류를 삽입하여 맞춤으로 발생할 수 있는 오류를 테스트하는 테스트 케이스를 선정하는 기법을 제안하였다. 이것은 컴포넌트 맞춤에 효과적인 테스트 케이스를 선정한다는 장점을 갖는 기법이다. '오류삽입 기법'과 '뮤테이션 테스트 기법'을 사용하여 컴포넌트 맞춤 테스트에 적합한 테스트 케이스를 추출하여 적합한 컴포넌트 기반 소프트웨어를 위한 테스트 기법을 개발한다.

본 기법이 실제 컴포넌트 아키텍처에 적용할 수 있도록 실제 연구 과제인 "데이터 품질 제어기술(DAQUIM)[5]"을 컴포넌트로 하여 샤모아 프레임 워크(Chamois Framework)에 적용하여 수정한 컴포넌트 맞춤 테스트 기법을 제안한다.

본 논문은 2장에서 관련연구를 기술하고, 3장에서는 COM 컴포넌트 맞춤 테스트 기법을 제안한다. 4장에서는 샤모아 프레임 워크 내에 적용한 사례를 보이며, 5장에서는 본 논문의 분석에 대해 기술하고, 6장에서는 본 논문의 결론 및 향후 연구 과제를 제시한다.

2. 관련연구

2.1 컴포넌트 맞춤 패턴

논문[2,3]은 컴포넌트 맞춤 패턴을 크게 '컴포넌트 맞춤 구문(syntactic) 패턴'과 '컴포넌트 맞춤 의미(semantic) 패턴'으로 나누어 제안하였다. 컴포넌트 맞춤 구문 패턴은 화이트박스 클래스를 구문상 '어떻게' 컴포넌트 맞춤을 하는지에 대한 패턴으로서 표1처럼 세 가지 패턴으로 나누어진다.

컴포넌트 맞춤 의미 패턴은 컴포넌트를 '왜' 컴포넌트 맞춤을 하는지에 대한 패턴이다. 컴포넌트 맞춤은 의미에 따라 컴포넌트 '속성변경 맞춤'과 '기능추가 맞춤'으로 구분된다. 이들을 컴포넌트 맞춤 의미 패턴으로 표2와 같이 정의하였다.

컴포넌트 맞춤 구문 패턴	설명
	주어진 W를 그대로 유지하고 W의 애틀리뷰트와 메소드에 대한 소스코드를 수정
	맞춤코드를 새로운 클래스 New로 작성하고 작성된 새로운 클래스를 주어진 W가 참조할 수 있도록 연관관계 설정
	맞춤코드를 새로운 클래스 New로 작성하고 작성된 새로운 클래스를 주어진 W가 이용할 수 있도록 새로 작성한 클래스를 주어진 화이트박스의 상위 클래스로 상속 상속관계 설정

표 1 컴포넌트 맞춤 구문 패턴

컴포넌트 맞춤 의미 패턴	설명
속성변경 맞춤	컴포넌트의 속성변수의 값을 변경시키는 맞춤
기능추가 맞춤	새로운 기능을 갖는 코드나 플러그인을 추가하는 맞춤

표 2 컴포넌트 맞춤 의미 패턴

2.2 Chamois Framework

샤모아[4]는 이화여자대학교 과학기술대학원에서 수행중인 IKEA(Integrated Knowledge Engineering Architecture) 프로젝트로써, 컴포넌트 기반 지식공학 프레임 워크이다. 샤모아 프레임 워크는 본 대학원에서의 여러 연구 전체를 하나로 묶어 서로의 연구가 상호 시너지 효과를 낼 수 있도록 하기 위한 것이다. 이렇게 함으로써 기존의 상업적인 지식공학 프레임 워크의 의미보다 더 큰 규모의 독특한 컴포넌트를 지니고 있는 새로운 지식공학 프레임 워크를 구축한다. 그림1은 전체적인 샤모아 프레임 워크를 보여주고 있다.

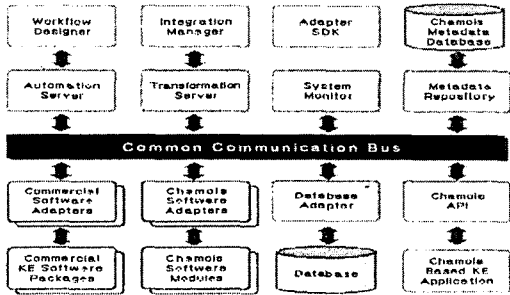


그림 1 샤모아 프레임 워크(Chamois Framework)

샤모아에서 개발한 컴포넌트 기술들은 다음과 같은 내용을 포함하고 있다.

- > Component-based software design, development, and testing
- > Data warehousing and metadata management
- > OLAP and data/text/Web mining
- > XML server, querying, mining
- > Networking
- > Security
- > E-commerce architecture, e-payment, personalization, m-commerce, voice-based access to the Internet

이를 위해, 개발된 모듈들은 COM, Web Service등과 같은 컴포넌트 아키텍처 기술들을 사용하였다.

3. COM 컴포넌트 맞춤 테스트 기법

컴포넌트는 공개되는 인터페이스와 공개되지 않는 핵심기능으로 구현된다. 컴포넌트 아키텍처 중의 하나인 COM도 컴포넌트의 핵심 기능 부분은 private나 protect로 구현하고, 인터페이스 부분은 public으로 구현하여 각각의 특성을 반영한다.

정의 1. 블랙박스 클래스 (B)

블랙박스 클래스란 컴포넌트 핵심기능 부분으로서 컴포넌트 사용자가 수정할 수 없는 부분이다. 이를 B로 표현한다.

■ 예 : " DAQUM" 컴포넌트의 B는 그림2의 B와 같다.

정의 2. 화이트박스 클래스 (W)

화이트박스 클래스란 컴포넌트 인터페이스 부분으로서 컴포넌트 사용자가 이를 수정하여 컴포넌트를 맞춤 할 수 있도록 하는 부분이다. 이를 W로 표현한다.

■ 예 : " DAQUM" 컴포넌트의 W는 그림2의 W와 같다.

정의 3. 컴포넌트 (BW)

컴포넌트는 B와 W가 결합된 단위로서 BW로 표현한다. 컴포넌트 맞춤으로 W가 변형된 BW를 cBW라고 하고, 오류가 삽입된 cBW를 fBW로 표현한다.

■ 예 : " DAQUM" 컴포넌트의 BW는 그림2와 같다.

```

BW
Public DAQUIM As Integer
Public PMODE As String
Public connmd As New ADODB.Connection
Public Function SetProperties()
    Form_SetProperties.PMODE = "1"
    Form_SetProperties.Show vbModal
    SetProperties = Form_SetProperties.DAQUIM
End Function
Public Function UpdateProperties(ByVal P_DAQUIM As Integer)
    Form_SetProperties.DAQUIMID=P_DAQUIM
    Form_SetProperties.PMODE = "1"
    Form_SetProperties.Show vbModal
End Function
Public Function Execute(ByVal P_DAQUIM As Integer)
    Call ProcessDAQUIM(P_DAQUIM)
End Function
Public Function ShowResult(ByVal P_DAQUIM As Integer)
    Form_ShowResult.DAQUIMID = P_DAQUIM
    Form_ShowResult.Show vbModal
End Function
Public Sub UpdateProperties(ByVal P_DAQUIM As Integer)
Public Sub InsertProperties()
Public Sub ShowGraph()
Public Sub Form_Load()
Private Const ConStr As String =
    "dsn=chamoisuid+sa.pwd"
Private Function ProcessDAQUIM(ByVal P_DAQUIM As Integer)
Private Sub Button_Cancel1_Click()
Private Sub Button_OK1_Click()
Private Sub DataLib_Button_Click()
Private Sub Button_Cancel3_Click()
Private Sub Button_OK3_Click()
Private Sub Form_Load()
Private Sub TreeView1_NodeClick(ByVal node As Node)
Private Sub ShowClear()
Private Sub HFG1_Click()
Private Sub Insert1_Button_Click()
Private Sub Edit1_Button_Click()
Private Sub Clear1_Button_Click()
Private Sub Delete1_Button_Click()
Private Sub DTYButton_Click()
Private Sub CTYButton_Click()
Private Sub TotGraphButton_Click()
Private Sub Form_Load()
Private Sub GetIDTYResult()
Private Sub CLOSE1_Button_Click()
Private Sub Form_Load()
Private Sub GetCTYResult()
Private Sub CLOSE2_Button_Click()
Private Sub Form_Load()
Private Sub CLOSE3_Button_Click()
    
```

그림 2 DAQUM 컴포넌트의 B, W, BW

3.1 COM 컴포넌트 맞춤 패턴

일반적으로 정의된 논문[2]의 컴포넌트 맞춤 패턴과 COM스펙 [6]을 참조하여, 다음의 2개 COM 컴포넌트의 맞춤 패턴을 정의한다.

(1) COM 컴포넌트 맞춤 패턴 1

다음은 COM 컴포넌트의 속성값을 변경하는 맞춤 패턴이다.

(2) COM 컴포넌트 맞춤 패턴 2

다음은 COM 컴포넌트의 속성값을 정의하는 제약사항을 변경하는 맞춤 패턴이다.

따라서, COM 컴포넌트 맞춤 패턴 1과 2는 컴포넌트 개발자에 의해 제공되는 내용을 사용자가 단지 선택하거나 설정하는 수준의 맞춤 패턴이므로, 컴포넌트 맞춤 패턴의 Pattern.syn1과 Pattern.sem1에 해당한다.

3.2 COM 컴포넌트의 오류삽입 대상과 오류삽입 연산자

오류삽입 대상(FIT)은 cW의 코드를 가운데 오류가 삽입되는 곳이다. 오류삽입 대상은 단순히 오류를 cW전체 또는 맞춤을 위해 수정된 코드, 즉, 맞춤코드 이외의 곳에 존재할 수 있다. 따라서, 컴포넌트 맞춤 패턴과 cW코드 가운데 B와의 상호작용에 따라 체계적으로 선정되는 것으로서 단순히 맞춤코드를 기준으로 오류삽입 대상이 선정되는 것이 아니라는 것을 알 수 있다.

오류삽입 연산자(FIO)는 cBW의 오류삽입 대상에 오류를 삽입하

는 연산자이다. 오류삽입은 fBW에서 문법적 오류가 발생하지 않아야 한다.

3.3 테스트 케이스 선정

컴포넌트 맞춤 패턴에 따른 오류삽입 연산자를 사용하여 생성된 fBW와 테스트 대상이 되는 cBW를 차별화하는 테스트 케이스를 선정한다.

4. 사례연구

본 논문에서 개발한 COM 컴포넌트 맞춤 테스트 기법을 적용하는 예를 보인다.

4.1 대상 컴포넌트

샤모아 프레임 워크가 최종 사용자에게 제공하는 데이터나 지식의 품질 측정 및 제어의 기준이 될 수 있는 DAQUM 컴포넌트[5]를 대상 컴포넌트로 한다. 그림2는 DAQUM 컴포넌트의 코드이다. B와 W의 각 메소드들의 소스코드는 지면 관계상 생략하고 각 메소드들의 시그너처만 표현하였다.

4.2 COM 컴포넌트 맞춤 테스트

DAQUM 컴포넌트는 지식공학 시스템 내에서 오류 데이터의 분류(Kim's et al분류에 의한 33개)에 따른 데이터 품질 측정을 한다. 본 예제에서는 품질등급을 제도화함에 따라 데이터 품질을 개선하고, 지식공학 시스템 내에서 얻어지는 최상의 결과를 얻을 수 있도록 컴포넌트 맞춤을 한다. 그림3은 cW로서 W를 컴포넌트 맞춤을 하여 변형된 코드이다. cW의 코드는 지면 관계상 변형된 코드부분만 보이고, 나머지 소스코드는 각 메소드들 시그너처만 표현하였다.

```
Public DAQUMID As Integer
Public PMode As String
Public connmd As New ADODB.Connection
Public Function SetProperties()
Public Function UpdateProperties(ByVal P_DaqumID As Integer)
Public Function Execute(ByVal P_DaqumID As Integer)
Public Function ShowResult(ByVal P_DaqumID As Integer)
Public Sub ShowGraph()
.....
If AdoRs.Fields(2) >= 75 Then
    L_degree.Caption = "A (First Degree)"
ElseIf AdoRs.Fields(2) <= 75 Or AdoRs.Fields(2) > 50 Then
    L_degree.Caption = "B (Second Degree)"
ElseIf AdoRs.Fields(2) < 50 Then
    L_degree.Caption = "C (Third Degree)"
End If
.....
Public Sub Form_Load()
Public Sub UpdateProperties(ByVal P_DaqumID As Integer)
Public Sub InsertProperties()
```

그림 3 DAQUM 컴포넌트의 맞춤후의 코드 (cW)

CBSD에서 DAQUM 컴포넌트를 재사용하여 컴포넌트 기반 소프트웨어를 개발할 때 발생하는 오류. 즉, 맞춤된 컴포넌트인 cBW에 존재하는 오류를 테스트하기 위해 본 기법을 적용한다.

■ 오류삽입 대상 : FIT

COM맞춤 패턴1에 의해 컴포넌트 맞춤된 DAQUM 컴포넌트의 FIT는 WPR부분인 ' B (Second Degree)' 이다. WPR부분에 오류를 삽입하여 테스트 케이스를 선정한다.

■ 오류삽입 연산자 : FIO

WPR부분에 WPRR을 FIO에 적용하여 테스트 케이스를 선정한다. WPRR연산자에 의해 cW에서 WPR부분인 ' B (Second Degree)' 이 fW에서 ' C (Third Degree)' 로 대체되었다.

■ 테스트 케이스 : TC

본 예제에서 cBW(x)=fBW(x)를 만족하는 x= " Chamois" , " Emp" , " Title" , " preference according to age" 을 선정하면, cBW에 적용한 결과가 Consider Purpose Data Quality : 85%, Not Consider Purpose Data Quality : 80.96%, Total Data Quality : 68.82%, Data Quality Degree : B (Second Degree)이고, fBW에 적용한 결과는 Consider Purpose Data Quality : 85%, Not Consider

Purpose Data Quality : 80.96%, Total Data Quality : 68.82%, Data Quality Degree : C (Third Degree)이다.

5. 분석

테스트 케이스를 cBW에 적용한 결과와 그에 대한 컴포넌트 사용자의 기대 결과가 다르다면 수행한 맞춤에 오류가 있는 것이다. 선정된 테스트 케이스 가운데 " Chamois" , " Emp" , " Title" , " preference according to age" 을 임의로 선택하여 cBW에 적용한 결과는 Consider Purpose Data Quality : 85%, Not Consider Purpose Data Quality : 80.96%, Total Data Quality : 68.82%, Data Quality Degree : B (Second Degree)이었고, 컴포넌트 사용자가 cBW에 기대하는 결과는 Consider Purpose Data Quality : 85%, Not Consider Purpose Data Quality : 80.96%, Total Data Quality : 68.82%, Data Quality Degree : C (Third Degree)이다. 따라서, 컴포넌트 사용자가 맞춤을 수행한 cBW에 오류가 있음을 감지할 수 있다.

6. 결론 및 향후 연구 과제

본 논문에서는 COM 컴포넌트를 적용하여, 실제 CBSD환경에 본 기법이 적용될 수 있음을 보였다. 본 기법이 실제 연구 과제인 "데이터 품질 제어기술(DAQUM)[5]"을 컴포넌트로 하여 샤모아 프레임 워크에 적용하기 위해 COM을 위한 FIT와 FIO를 선정하여 COM에서의 테스트 사례를 보였다. 그리고, COM 컴포넌트의 인터페이스 가운데 맞춤 오류가 일어나는 곳에만 오류를 삽입하여, 맞추어진 컴포넌트와 오류가 삽입된 컴포넌트를 차별화하는 테스트 케이스를 선정하였다.

향후, 본 기법의 컴포넌트 오류를 발견하는 능력에 대한 실험을 수행하여 본 기법의 테스트 케이스가 갖는 컴포넌트 맞춤 오류에 대한 적정성을 평가할 계획이다. 이를 위해 본 기법의 FIT와 FIO를 적용하여 fBW와 그에 대한 테스트 케이스를 자동 생성하는 도구를 구현할 계획이다.

참고문헌

[1]Hojjin Yoon, Byoungju Choi, " Testing in CBSD," Software Engineering Lettr," 1999.
 [2]Hojjin Yoon, Byoungju Choi, " A Test Technique for Component Customization Failure," Journal of Korea Information Science Society, 27(2):148-156, 2000.
 [3]Hojjin Yoon, Byoungju Choi, "Component Customization Testing Technique Using Fault Injection Technique and Mutation Test Criteria," in Proc. Mutation 2000, San Jose, USA, pp 93-100, Oct.6-7, 2000.
 [4]Won Kim, Ki-Joon Chae, Dong-Sub Cho, Byoungju Choi, Anmo Jeong, Myung Kim, KiHo Lee, Meejeong Lee, Sang-Ho Lee, Seung-Soo Park, Hwan-Seung Yong, " The Chamois Component-based Knowledge Engineering Framework" , IEEE Computer Journal, May 2002.
 [5]양자영, 최병주, "소프트웨어 사용자 관점의 데이터 품질 측정 방안," 제28회 정보과학회 추계학술대회, pp436-438, Oct 19-20, 2001.
 [6]Microsoft, Component Object Model Specification 0.9. at URL: <http://microsoft.com>
 [7]John Cheesman and John Daniels, " UML Component : A Simple Process for Specifying Component-Based Software," Addison-Wesley,2000.